List recommendations with multi-armed bandits

Camille-Sovanneary Gauthier

UNIVERSITÉ DE RENNES 1

THÈSE DE DOCTORAT DE

L'UNIVERSITÉ DE RENNES 1

ÉCOLE DOCTORALE N° 601
Mathématiques et Sciences et Technologies de l'Information et de la Communication
Spécialité : *Informatique*

Par
Camille-Sovanneary GAUTHIER

List recommendations with multi-armed bandits

Thèse présentée et soutenue à Rennes, le 17 mars 2022
Unité de recherche : IRISA

Rapporteurs avant soutenance :

Vianney PERCHET Professeur, ENSAE
Philippe PREUX Professeur, Université de Lille

Composition du Jury :

Président :	François TAÏANI	Professeur, Université Rennes 1
Examinateurs :	Audrey DURAND	Assistant Professor, Université Laval
	Jeremie MARY	Senior Researcher, Criteo
	Claire VERNADE	Research Scientist, DeepMind
Dir. de thèse :	Elisa FROMONT	Professeure, Université Rennes 1, IRISA
Co-dir. de thèse :	Romaric GAUDEL	Maitre de conférence, ENSAI, CREST

Invité(s) :

Bruno GUILBOT Head of Data and Artificial Intelligence, Louis Vuitton

Acknowledgement

Je ne suis pas une personne très manuelle... C'est un fait que mes proches et moi-même avons constaté assez tôt. Par contre, l'art et l'artisanat, qui nécessitent de fortes compétences manuelles pour se matérialiser, me fascinent. Ce sont des activités créatives, passionnées et passionnantes et imprégnées de partages et de transmissions. Je trouve cela beau. Mais bon, ce n'était pas dans mes cordes et j'ai suivi un cursus scolaire plus traditionnel pour notre époque. En plus, je préfère les mathématiques et l'informatique. Adieu cafés d'artistes, bonjour pragmatisme ! C'est en tout cas ce que je pensais.

Maintenant que je termine ma thèse, que je regarde ce qui s'est passé et ce qui m'attend, je me vois en artisane du monde numérique, à façonner les interactions de demain. Ce regard nouveau, j'ai pu le forger avec l'aide de toutes les personnes sur lesquelles j'ai pu compter et que je veux prendre le temps de remercier ici. Parce que comme toute artisane, avec cette thèse, j'ai pu débuter un compagnonnage riche de rencontres.

Je tiens tout d'abord à remercier mon directeur et ma directrice de thèse, mes mentors dans cet art qu'est la recherche : Romaric Gaudel et Elisa Fromont. Je vous remercie pour votre temps, votre patience et tous vos conseils. Merci Elisa pour ton énergie et ton enthousiasme. Tu m'as donné de multiples clés et cartes pour explorer le monde de la recherche et décrypter ses codes. Ton accompagnement m'a été précieux pour prendre la hauteur nécessaire et tes critiques constructives m'ont permis d'aiguiser ma clarté et ma pédagogie, qui sont des compétences qui me sont maintenant indispensables. Merci pour ton rire et tes encouragements qui ont illuminé ces trois années. Merci Romaric pour cette discussion, alors que j'étais encore élève à l'ENSAI, qui m'a confortée dans le fait qu'une thèse était un beau projet auquel je pouvais m'accrocher. Merci ensuite d'avoir construit ce sujet avec Bruno et moi. Merci d'avoir cru en ce projet et en mes capacités. Merci enfin de m'avoir poussé à aller toujours plus loin dans des nouveaux domaines et de nouveaux axes de travail. Ta passion pour la recherche et ton intérêt pour tous les sujets me fascinent. Merci de m'avoir fait autant évoluer pendant ces trois années aussi bien humainement que techniquement. Même si j'ai vécu cette thèse physiquement loin de vous, je me suis toujours sentie soutenue et écoutée. Je me sens chanceuse et reconnaissante de vous avoir eus tous les deux comme directeur et directrice de thèse et d'avoir partagé ces années avec

vous.

Je ne sais pas si lors d'un "vrai" compagnonnage le travail réalisé au cours de ce parcours initiatique est jugé par des paires. En thèse, oui. Et pour leurs remarques et questions constructives, je tenais à remercier mes rapporteurs, Philippe Preux et Vianney Perchet, ainsi que mon jury : Audrey Durand, Claire Vernade et Jeremie Mary. Je tenais à remercier également François Taïani qui en plus d'avoir présidé mon jury, a vérifié le bon déroulement de cette thèse au côté de Charlotte Laclau que je remercie également. Leur implication dans mon Comité de suivi a permis de poser des balises concrètes et d'une réelle aide pour moi dans cette thèse.

Pendant ces trois ans, j'ai pu travailler simultanément dans deux ateliers rennais au côté de maîtres de disciplines variées et de leurs "apprentis".

L'atelier ensaien qui m'avait initiée et donné mes premiers sujets de composition en apprentissage automatique, a continué de me donner des "modèles" pour mes œuvres. Continuer dans cet atelier m'a permis de (re)découvrir des artistes des statistiques et de profiter de la sagesse d'Amandine D., des conseils et de la solidarité de Steven G., de la fantaisie d'Edouard G., du bon sens de Max D., de la force tranquille d'Eli C., de la vitalité de Daphné A., de l'humour de Sunny W. et de la gentillesse de Guillaume Fl., Hassan M., Guillaume Fr. et Camille M.

L'atelier Lacodamien a été l'occasion de découvrir de nouvelles applications et une nouvelle manière de penser et d'utiliser l'informatique. Et cette découverte, j'ai pu la faire grâce à des artistes d'une bienveillance absolue tels que Alexandre T., Laurence R., Véronique M., Christine L., Tassadit B., Peggy C., Gaelle T. et Luis G.(qui m'a aussi fait découvrir le chocolat équatorien !). J'ai découvert aussi de nombreuses applications et nouvelles pistes grâce aux "petits" dont je faisais partie quelques jours par mois au côté de Johanne B. (merci de m'avoir hébergée et d'avoir animé mes passages à Rennes), de Gregory M. (merci de m'avoir initiée à la politique), de Yichang W. (merci pour ton sourire communicatif), de Heng Z. (merci pour tes conseils sur IGRIDA), de Mael G. (merci de m'avoir introduite à vos soirées jeu), de Colin L. (merci de m'avoir fait découvrir le reggae nu-roots), de Antonin V. (merci de m'avoir mise en garde sur toutes ces cyberattaques), de Lenaig C. (merci d'être aussi compréhensive face à mes reviews douteuses et mes répétitions bancales), de Julien D. et Simon C. ("merci" de m'avoir jugée pour mes goûts gastronomiques), de Josie S., d' Olivier G., de Nassim A. et de beaucoup d'autres...

Etant en thèse CIFRE et dans cette métaphore artistique, certains pourraient penser

que mon entreprise, Louis Vuitton, a tenu le rôle de mécène. Mais le département Innovation Digitale dans lequel j'étais intégrée a été bien plus pour moi. Cette équipe est comme un grand collectif d'artistes et d'artisans mettant en œuvre leur savoir-faire pour constuire des projets innovants et variés, et répondre ainsi aux besoins de LV et de ses clients : architectes de projets, poéte(sse)s de la veille, ébénistes du code, sculpteur(trice)s de données...

Je tenais donc à remercier Bruno Guilbot, Agnès Vissoud et Eliot Barril pour leur confiance et pour avoir permis à cette thèse de voir le jour. Un merci particulier à Bruno pour ton soutien envers ce projet et ton aide précieuse dans sa mise en place. Merci d'être un tel initiateur et moteur pour nos projets IA même lorsqu'ils sont (trop) disruptifs et pour ton enthousiasme et ta bienveillance. Merci également à toute l'équipe pour votre curiosité et votre soif de partage qui ont coloré ces trois années. Merci à Julie B. pour ta belle et bonne humeur contagieuse et tous les moments de partage dont tu es souvent l'initiatrice, merci à Badr S. d'avoir été un binôme de choc et pour continuer d'être aussi partant et coopératif pour nos projets, merci à Laura E., Anne P., Gordan G. et Félina R. de me faire découvrir toutes ces dernières tendances (mode, innovation et viennoiseries), merci à Julien L. et à Pierre L. de m'avoir fait découvrir nombre de jeux, séries et aussi méthodes de traitement stat et data, merci à Léa F. d'avoir relevé le defi d'un stage sur les bandits contextuels, merci à Fabien S., Ken B., Basile M., Amelie N., Thomas P., Matthieu B., Charlotte P., Maud L., Charlène D., Florent C., Aurélien L., Thibault P., Jordan F., Steven L., Nathan V., Yuan X., Aymeric F., Ming Z. et tous ceux que je n'ai pas pu citer pour ces moments passés ensemble au Bailleul, écrin de tant d'œuvres fantastiques.

Mes cafés d'artistes ont été des cafet d'écoles et de labo, mais aussi des écoles d'étés et des conférences (même virtuelles). A ces occasions, on écoute sur scène les maîtres, on refait le monde avec d'autres passionnés et surtout, on se soutient. Merci à Guillaume A. de l'avoir fait aussi autour de pâtisseries et à Dorothé K. de l'avoir fait au travers de l'expérience MT180.

Enfin, j'ai la chance d'être soutenue par une famille et des amis formidables. Merci à mes parents pour tout et notamment de m'avoir appris que je pouvais faire tout ce que je voulais si j'y mettais les efforts et l'intégrité nécessaires. Merci à ma petite soeur Eva-Kalyane, qui m'émerveille par sa force de caractère et qui me pousse à donner le meilleur en acceptant mes faiblesses. Merci à ma moitié, mon pilier ces dernières années qui m'a soutenue, et même s'il ne comprenait pas pourquoi je me lançais dans "une telle galère",

m'a permis de voir toujours le positif.

Je remercie tous mes proches qui sont présents dans les bons comme dans les moments difficiles et notamment Sao. Merci à mes amis que je connais depuis le collège, le lycée, la prépa ou l'ENSAI et qui, même si nos chemins ont divergé géographiquement ou professionnellement, sont toujours présents et me soutiennent.

Enfin merci à mon grand-père, Marcel Gauthier, qui a vu mes premiers déboires de thèse et qui aurait adoré m'entendre parler de mes publications, qui aurait suivi ma soutenance avec tout l'intérêt du monde et qui aurait lu ce mémoire de la première à la dernière page. Merci de m'avoir montré que la curiosté était une force et que le mot 'pourquoi' était irremplaçable.

"Ce qui est important, ce n'est pas de finir une œuvre, mais d'entrevoir qu'elle permette un jour de commencer quelque chose." ***Joan Miro***

Ça tombe bien, la recherche est une œuvre infinie.

Table of Contents

Long summary (in French) **11**

Introduction **19**

Recommendation systems and Louis Vuitton 19

Online learning to rank at Louis Vuitton 22

Thesis outline 24

1 Background on bandit-based recommender systems **27**

1.1 User click behavioral models 28

1.1.1 Position-based model 29

1.1.2 Cascading model 30

1.1.3 Others click behavioral models 31

1.2 Bandit algorithms 31

1.2.1 Generality 32

1.2.2 Thompson sampling 33

1.2.3 Upper confidence bound algorithm 34

1.2.4 Combinatorial bandits 35

1.3 My thesis setting: learning to rank in a semi-bandit setting 36

1.3.1 Bandits for click behavioral model 36

1.3.2 Performance evaluation 37

1.3.3 Choice of the environment to evaluate bandit algorithms 38

1.3.4 Datasets 39

1.4 Conclusion 42

2 Related work **43**

2.1 Bandits on PBM 43

2.1.1 PMED 44

2.1.2 Focus on (KL)CombUCB1 45

2.2 Bandits on other click behavioral models 47

2.2.1 Focus on TopRank 47

2.3 Related algorithms . . . 49
2.3.1 Focus on OSUB . . . 49
2.4 Conclusion . . . 51

3 Unimodal bandit for PBM **53**
3.1 Relation with unimodality . . . 54
3.2 Parametric graph for unimodal ranking bandit . . . 57
3.3 Theoretical analysis . . . 59
3.3.1 Discussion . . . 63
3.4 Practical results . . . 65
3.5 Conclusion . . . 67

4 MCMC bandits for PBM **69**
4.1 Thompson sampling with approximation approaches . . . 70
4.1.1 Approximation based on Metropolis Hasting . . . 71
4.1.2 Approximation based on Langevin gradient descent . . . 76
4.1.3 Overall complexity . . . 78
4.2 Practical results . . . 78
4.3 Conclusion . . . 89

5 Unimodal bandits for other click behavioral models **91**
5.1 Model assumption . . . 93
5.2 UniRank: unimodal bandit algorithm for generic online ranking . . . 95
5.3 Theoretical analysis . . . 100
5.4 Practical results . . . 103
5.5 Conclusion . . . 105

Conclusion **107**
Take away . . . 107
Contribution for Louis Vuitton . . . 110
Echo chamber and exploration behavior . . . 111
Perspectives . . . 112

Bibliography **115**

Appendix **123**

A GRAB **124**

A.1 Notations . . . 124
A.2 Proof of Lemma 1 (PBM Fulfills Assumption 1) . . . 126
A.3 Preliminary to the Analysis of GRAB . . . 128
A.4 Proof of Theorem 2 (Upper-bound on the Regret of KL-CombUCB) . . . 128
A.5 Proof of Lemma 2 (Upper-bound on the Number of Iterations of GRAB for which $\tilde{\boldsymbol{a}}(t) = \tilde{\boldsymbol{a}} \neq \boldsymbol{a}^*$) . . . 131
A.6 Proof of Lemma 3 (Upper-bound on the Number of Iterations of GRAB for which $\tilde{\boldsymbol{\pi}}(t) \notin \Pi_{\boldsymbol{\rho}}(\tilde{\boldsymbol{a}})$) . . . 136
A.7 S-GRAB: OSUB on a Static Graph . . . 137

B UniRank **139**

B.1 Organisation of the Appendix . . . 139
B.2 Notations . . . 139
B.3 Proof of Lemma 4 (PBM and CM Fulfills Assumptions 2, 4, and 5) . . . 143
B.4 Technical Lemmas Required by the Proof of Theorem 3 . . . 145
B.4.1 Proof of Lemma 5 (Pseudo-Unimodality Assuming a Total Order on Items) . . . 146
B.4.2 Minimum Expected Click Difference . . . 146
B.4.3 Upper-bound on the Number of High Deviations for Variables with Lower-Bounded Mean . . . 147
B.4.4 Upper-Bound on the Number of Upper-Estimations of a Pessimistic Estimator . . . 152
B.5 Proof of Theorem 3 (Upper-Bound on the Regret of UniRank Assuming a Total Order on Items) . . . 154
B.5.1 Upper-Bound on the Number of Sub-Optimal Merges of UniRank when the Leader is the Optimal Partition . . . 155
B.5.2 Upper-Bound on the Expected Number of Iterations at which the Leader is not the Optimal Partition . . . 157
B.5.3 Final Step of the Proof of Theorem 3 (Upper-Bound on the Regret of UniRank Assuming a Total Order on Items) . . . 165
B.6 UniRank's Theoretical Results While Facing State-of-the-Art Click Models 169
B.6.1 Proof of Corollary 1 (Upper-Bound on the Regret of UniRank when Facing CM* Click Model) . . . 169

B.6.2 Proof of Corollary 2 (Upper-Bound on the Regret of UniRank when Facing PBM* Click Model) 170

Long summary (in French)

Cette thèse CIFRE s'inscrit dans le projet de Louis Vuitton d'améliorer l'expérience de ses clients sur l'ensemble de ses interfaces numériques.

En tant que marque de luxe, Louis Vuitton (LV) a toujours accordé une attention particulière aux services dispensés à ses clients afin d'offrir l'*"ultime expérience d'achat"*. En magasins, les conseillers clients sont les ambassadeurs de la "Maison" et sont formés spécifiquement pour faire de chaque visite des clients, une expérience satisfaisante.

La digitalisation croissante de la société amène toutes les marques, y compris les marques de luxe, à moderniser leurs processus de vente pour s'adapter et se fondre dans les nouveaux besoins de leurs clients. De plus, les pages web traditionnelles sont appelées à changer avec l'arrivée de l'incrustation de produits dans des vidéos ou via les nouvelles expériences 3D [1] qui permettront aux utilisateurs d'avoir des interactions plus immersives et plus de liberté dans leur navigation web. Ces interfaces ouvertes s'imposent comme une révolution incontournable pour les marques et vont creuser un fossé entre les marques s'adaptant à ce nouveau monde numérique et celles qui ne le font pas. Pour assurer l'accès à ces produits et services, Louis Vuitton dispose de différents canaux : son site internet, son chatbot, appelé Assistant Virtuel, son application " LV app " et l'utilisation d'applications sociales et mobiles tel que Instagram, Wechat ou Line.

Sur chaque canal, des équipes dédiées sont chargées de concevoir des *expériences* pour satisfaire les besoins des clients. Trois exemples d'expériences proposées sur le site e-commerce de LV sont présentées sur la Figure 4 : la page d'accueil dédiée à la mise en avant des actualités LV, une grille de produits dédiée à la suggestion de cadeaux et une grille de "looks" pour présenter les produits de prêt-à-porter. L'assistant virtuel est également capable de répondre de manière engageante et interactive aux clients. La Figure 5 donne un exemple d'interaction entre le chatbot et un utilisateur pour découvrir les produits de la marque.

Néanmoins, les clients ont de grandes attentes vis-à-vis de l'expérience proposée par Louis Vuitton et, comme la part des ventes à distance est en augmentation, ces attentes

1. https://journalduluxe.fr/fr/mode/louis-vuitton-un-madison-square-garden-virtuel-pour-le-lancement-de-la-capsule-nba

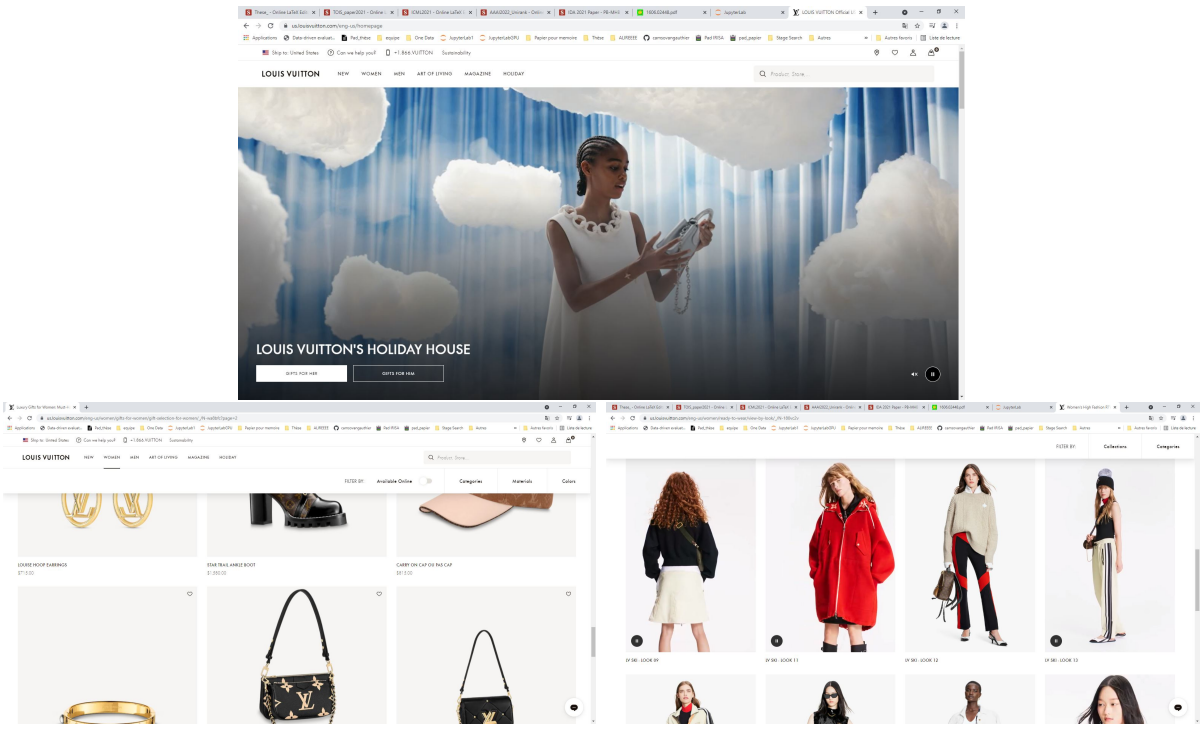


Figure 1: Pages du site e-commerce de Louis Vuitton. En haut : page d'accueil. En bas à gauche : la section "Cadeaux pour femmes" avec un affichage en grille. En bas à droite : la section "Prêt-à-porter par look", avec un affichage de plusieurs produits par look sur une grille.

ont influencé la conception et la gestion de tous les canaux de communication. Actuellement, sur le site e-commerce de Louis Vuitton, les recommandations sont basées sur des règles commerciales, sur de la similarité visuelle ou sur la popularité des produits et les actions passées des utilisateurs. Ces différentes stratégies de recommandations ont pour but de répondre aux besoins des clients en matière soit de conseils génériques, soit de recommandations alternatives ou encore pour pousser du contenu sur la marque. Comme de nouveaux produits sont fréquemment ajoutés au catalogue de produits LV, les recommandations sur le site e-commerce doivent faire preuve de dynamisme afin d'inclure efficacement ces nouveaux produits.

Les cas d'utilisation présentés précédement, tel que la construction de grille ou la découverte de produit grâce au chatbot ont en commun de choisir et d'ordonner K éléments à proposer à un utilisateur parmi L possibles. Cette configuration est connue sous le nom de recommandation de liste et sera le point central de cette thèse.

Les *systèmes de recommandations* (RS) sont conçus pour aider les utilisateurs à choisir

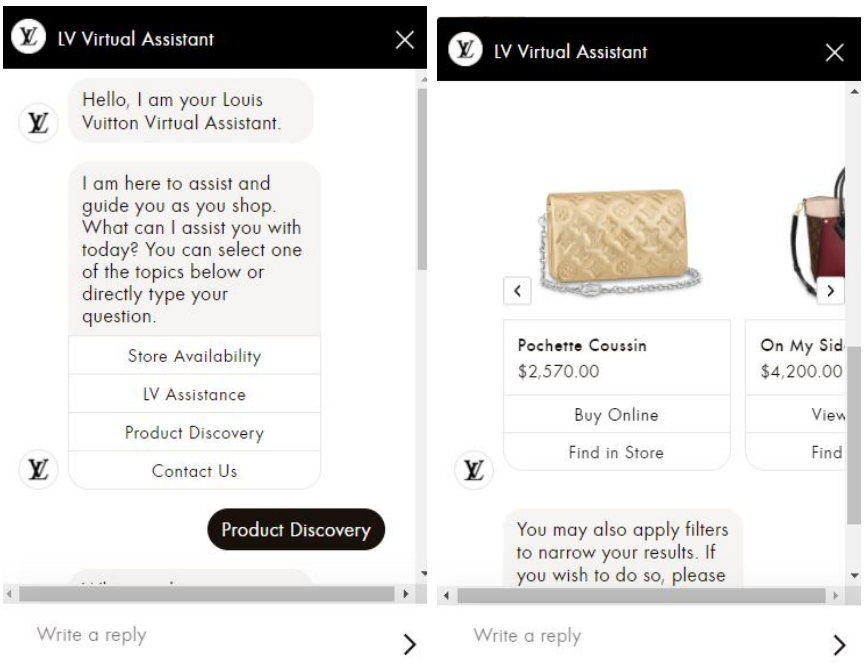


Figure 2: Product Discovery on Louis Vuitton's Virtual Assistant.

des éléments pertinents. Ces éléments peuvent être très divers (chansons, publicités, films ...) et sont souvent destinés à être affichés sur des pages web, parmi un très grand nombre d'éléments semblables. A la différence d'un conseiller de vente humain qui vous aurait promené dans tous les rayons d'un magasin physique, un système de recommandations peut aider simultanément des milliers d'internautes à rechercher des millions de produits. Ce processus automatique aide les entreprises à satisfaire leurs clients digitaux. La diversité des systèmes de recommandations et de leurs implémentations permet de répondre à différents besoins en termes d'interactions avec les clients : remplissage du panier, choix alternatifs, ... ils peuvent se concentrer sur une seule recommandation ou en fournir plusieurs à la fois pour donner plus de liberté aux utilisateurs. À chaque appel, ces systèmes sélectionnent K articles parmi L articles potentiels, $K \leq L$. Les retours des utilisateurs sont ensuite collectés pour chaque élément affiché, reflétant la pertinence des choix proposés : temps d'écoute, taux de clics, etc. Habituellement, ces retours sont utilisés simultanément lors d'une phase d'apprentissage statique (ou "batch") en appliquant par exemple des méthodes de filtrage collaboratif [57], des méthodes basées sur le contenu [47] ou en intégrant des caractéristiques décrivant les utilisateurs et les éléments [10, 65]. Cependant, les retours utilisés pour ces types d'apprentissages statiques ne sont disponibles que pour les éléments qui ont été effectivement présentés à l'utilisateur

jusqu'alors. L'apprentissage des modèles précédents influence donc les recommendations alors que pour les modèles classiques d'apprentissage statique, les données sont supposées indépendantes. Les *systèmes de recommandation en ligne* (ORS) sont développés pour surmonter ce problème de dépendance entre données. Le problème dit du *bandit manchot à K-bras* avec *retour semi-bandit* [19, 9] est une façon standard de décrire ce cadre : celui-ci tend à recommander itérativement un ou plusieurs éléments parmi un ensemble plus large d'éléments possibles, chacun d'entre eux étant indépendant des autres, puis il reçoit un retour pour chaque élément recommandé qui sera utilisé par l'algorithme pour choisir la prochaine liste de recommandations. Cette utilisation des retours amène l'algorithme à adopter deux types de comportement : d'une part, l'algorithme présente des articles avec peu ou pas de retours aux utilisateurs afin de collecter des informations sur tous les articles possibles et, d'autre part, l'algorithme promeut les articles qui ont les meilleurs retours.

Un autre problème, lié à l'ordonnancement, consiste à afficher les K éléments choisis aux bonnes positions pour maximiser l'attention de l'utilisateur. Des exemples typiques de tels affichages sont (i) une liste de nouvelles, visibles une par une ; (ii) une liste de produits, disposés par rangées ; ou (iii) des publicités réparties partout sur une page web. Plusieurs approches ont été proposées pour apprendre à positionner les meilleurs éléments aux meilleures positions [53, 15, 44]. Ces approches sont appelées *bandits manchots multi jeu* ou *apprentissage d'ordonnancement en ligne* (OLR).

Pour s'attaquer au problème d'apprentissage d'ordonnancement en ligne rencontré sur le site de LV, il faut comprendre et identifier le modèle de comportement de clics suivi par les utilisateurs de LV. Une fois le modele identifié, des algorithmes peuvent être développés pour estimer efficacement les paramètres de ce modèle.

Sur le site de Louis Vuitton, par exemple, les produits sont affichés sur des grilles de différents formats, en fonction de l'appareil utilisé (téléphone, ordinateur...). Ces différents types d'affichage entraînent des sens de lecture variant selon les utilisateurs. Ainsi, il est important de comprendre comment les clients interagissent avec les recommandations affichées car l'attention des clients envers un article affiché est impactée par sa position sur la grille. En plaçant les produits adéquats dans des positions qui seront vues en premier par les clients, ils trouveront plus facilement ce dont ils ont besoin et leur expérience sera améliorée. Cela introduit un nouveau défi qui consiste à ordonner les articles et les positions.

Un client peut donner différent retours lorsqu'on lui présente une liste de recomman-

dations, c'est-à-dire une liste ordonnée de K éléments. Ces retours peuvent être un clic unique sur le premier élément pertinent ou le temps passé à consulter une recommandation. Nous considérons ici qu'un retour est une liste de K booléens (clic ou non clic), un pour chaque élément présenté. Les modèles comportementaux de clics visent à fournir un modèle paramétrique des interactions entre les clients et les listes de recommandations. Ce modèle définit les probabilités de clic pour chaque élément d'une liste. De nombreux modèles comportementaux de clics permettent de comprendre comment évolue l'attention (partielle) des clients en fonction des positions [55, 16]. La question principale est de comprendre si un élément situé à une position donnée a été vu ou non pour ensuite déterminer si un élément ne reçoit pas de clic parce qu'il n'était pas pertinent ou parce qu'il n'a pas été vu. Cette probabilité de clic dépend à la fois de la pertinence d'un élément et de l'impact de sa position. Les différents modèles de comportement de clics existants mettent en œuvre cette hypothèse de manière différente : le modèle basé sur la position (PBM) suppose que la pertinence d'un élément et l'impact de sa position sont indépendants ; le modèle en cascade (CM) suppose que les clients examinent les positions de haut en bas itérativement.

Les produits étant proposés sous la forme d'une grille sur le site de LV, la lecture d'une page n'est pas intuitive/conventionnée : il peut exister différents sens de lecture. Ainsi, plus que sélectionner les produits les plus pertinents, il s'agit de savoir aussi les positionner dans cette grille pour optimiser la visite des utilisateurs. Dans ce genre de situations, plusieurs modèles de comportement ont été identifiés : le modèle PBM semble le plus adapté du fait de l'indépendance des impacts des positions et des items, qui facilite l'adaptation aux différents sens de lecture.

Une fois le modèle comportemental de clics le plus pertinent identifié, les paramètres doivent être déduits à l'aide d'un algorithme efficace. Le site e-commerce de LV attire chaque jour de nombreux visiteurs, et met en avant le catalogue de produits de LV, qui contient des dizaines de milliers d'articles et est fréquemment renouvelé. En outre, l'affichage du site de e-commerce est essentiellement statique pour l'instant, avec quelques changements lorsque de nouvelles règles commerciales sont introduites. Tout cela conduit à très peu d'interactions par produit et par client, et ces interactions se concentrent sur les mêmes produits. Cette situation est un des inconvénient des approches traditionnelles de recommandation telles que le filtrage collaboratif et la factorisation matricielle qui ne peuvent pas avoir de bonnes estimations pour les produits peu mis en avant. Pour surmonter cette concentration sur peu d'éléments, les algorithmes de bandits manchots peuvent être

mis en œuvre en complément des systèmes de recommandations existants. Grâce à leur apprentissage en continu, les algorithmes de bandits manchots rendent l'affichage plus dynamique et apprennent sur tous les produits grâce aux interactions avec les utilisateurs et au mécanisme d'exploration des bandits manchots. Comme K articles sont affichés simultanément et que les interactions sont recueillies sur chaque article, nous sommes confrontés à ce que l'on appelle le retour semi-bandit à jeux multiples.

La structure de cette thèse est la suivante. Le chapitre 1 présente les idées générales sur les deux principaux concepts étudiés dans cette thèse : les modèles comportementaux de clics et les algorithmes de bandits manchots. Différents modèles de clics sont décrits (Section 1.1) à la fois en termes d'interaction avec l'utilisateur mais aussi selon des hypothèses générales sur la définition de la probabilité des clics. Un cadre général pour les algorithmes de bandits et quelques détails sur les algorithmes standards tels que Thompson Sampling et UCB sont présentés dans la section 1.2. La section 1.2.4 se concentre sur les bandits manchots combinatoires qui s'appliquent de façon similaire à certaines de nos contributions, notamment sur la manière d'associer des produits à des positions, qui peut être considérée comme une recommandation combinatoire. Pour conclure ce chapitre, la section 1.3 présente le cadre utilisé dans cette thèse. L'accent est mis sur l'évaluation en définissant les mesures de performance, le processus de génération des retours utilisateurs et les jeux de données utilisés.

Le chapitre 2 présente des algorithmes de l'état de l'art liés au contexte de la thèse présenté dans le chapitre précédent et aux contributions présentées dans les chapitres suivants. Les algorithmes sont divisés en trois groupes : les algorithmes spécialisés dans le modèle basé sur les positions (PBM), les algorithmes conçus pour d'autres modèles de clics ou pour les cas où les modèles de clics ne sont pas spécifiés et enfin les algorithmes dont la structure est proche des contributions présentées dans cette thèse. Au sein de ces groupes, une attention particulière est portée aux principaux algorithmes de l'état de l'art.

Le chapitre 3 introduit une contribution basée sur les bandits manchots unimodaux, publiée dans [26] et présentée à [25]. Cette adaptation du cadre des bandits manchots unimodaux au cas PBM est présentée dans la section 3.1. Ensuite, notre algorithme GRAB est présenté dans la section 3.2. L'analyse théorique et les résultats empiriques sont présentés dans les sections 3.3 et 3.4.

Le chapitre 4 présente une contribution concernant spécifiquement le problème PBM en couplant les bandits manchots par échantillonnage de Thompson et les méthodes

d'échantillonnage par approximation de Monte Carlo par chaîne de Markov. Ce travail a été publié dans [23] et présenté à [24]. Une version étendue de [23] est en cours de révision dans le journal *ACM Transactions on Information Systems*. La section 4.1 présente les deux méthodes d'approximation utilisées dans cette contribution ainsi que les algorithmes associés : PB-MHB (Position Based Metropolis-Hastings Bandit) et PB-LB (Position Based Langevin gradient Bandit). La section 4.2 donne des résultats empiriques et montre que, même si cette contribution n'a pas de preuve théorique, contrairement à celle présentée dans le chapitre précédent, elle fournit les algorithmes les plus performants.

Le chapitre 5 présente une contribution conçue pour recommander dans un cadre comportemental plus large que dans les deux chapitres précédents. Cette contribution adapte les bandits unimodaux pour s'adapter à plusieurs modèles de clics. Ces adaptations sont présentées dans les sections 5.1 et 5.2. L'analyse théorique et les résultats empiriques sont donnés dans les sections 5.3 et 5.4.

Pour conclure, nous faisons la synthèse de ces trois contributions et nous discutons de l'impact social de tels systèmes. Enfin, nous avançons quelques perspectives mises en lumière par ce travail, tant du point de vue de la recherche que du point de vue de l'entreprise Vuitton.

Publications

Voici la liste des articles rédigés dans le cadre de cette thèse :

— Publications en conférences internationales [23] and [26]:
  — *Camille-Sovanneary Gauthier, Romaric Gaudel, Elisa Fromont*, "Bandit Algorithm for both Unknown Best Position and Best Item Display on Web Pages", IDA: 19th International Symposium on Intelligent Data Analysis, Porto,Portugal, 2021.
  — *Camille-Sovanneary Gauthier, Romaric Gaudel, Elisa Fromont and Aser Boammani Lompo*, "Parametric Graph for Unimodal Ranking Bandit", ICML: Proceedings of the 38th International Conference on Machine Learning, virtual, 2021.
— Publications en conférences nationales [24] and [25]:
  — *Camille-Sovanneary Gauthier, Romaric Gaudel, Elisa Fromont*, "Bandits manchots avec échantillonnage de Thompson pour des recommandations basées sur les positions", CAp'2020 (Conférence d'Apprentissage), virtual, 2020.
  — *Camille-Sovanneary Gauthier, Romaric Gaudel, Elisa Fromont and Aser Boam-*

mani Lompo, "Ordonnancement d'objets par bandits unimodaux sur des graphes paramétriques", CAp'2021 (Conférence d'Apprentissage), Saint-Etienne, 2021.

— En révision (journal):
  — *Camille-Sovanneary Gauthier, Romaric Gaudel, Elisa Fromont*, "MCMC-based Thomson Sampling Algorithms for Online Recommendations in the Position-Based Model", ACM Transactions on Information Systems (TOIS).

— En soumission à ICML 2022 (refusé à NeurIPS 2021 et amélioré):
  — *Camille-Sovanneary Gauthier, Romaric Gaudel, Elisa Fromont*, "UniRank: Unimodal Bandit Algorithm for Online Ranking".

INTRODUCTION

This industrial thesis is part of Louis Vuitton's project to enhance its clients' experience on its digital interfaces.

Recommendation systems and Louis Vuitton

How Louis Vuitton advises its clients today

As a luxury brand, Louis Vuitton (LV) has always paid a particular attention to its client services in order to deliver the, so called, *ultimate premium experience*. Client Advisors are the ambassadors of the "Maison" in stores and are trained specifically to make every visit in the store, a satisfactory experience. Figure 3 shows a typical Louis Vuitton store where various products are displayed according to both specific LV guidelines and to the know-how of the store's crew.

The digitalization of everyone life brings luxury brands to be accessible online and adapt and blend in their clients' new needs. Moreover, traditional web pages are destined to change with the arrival of incrustation of product in videos and new 3D experiences [2] which will give users more immersive interactions and more liberty in their web browsing. These open interfaces will broaden the gap between brands which adapt to the digital

2. https://journalduluxe.fr/fr/mode/louis-vuitton-un-madison-square-garden-virtuel-pour-le-lancement-de-la-capsule-nba

Figure 3: Display of various products in Montaigne Store in Paris.

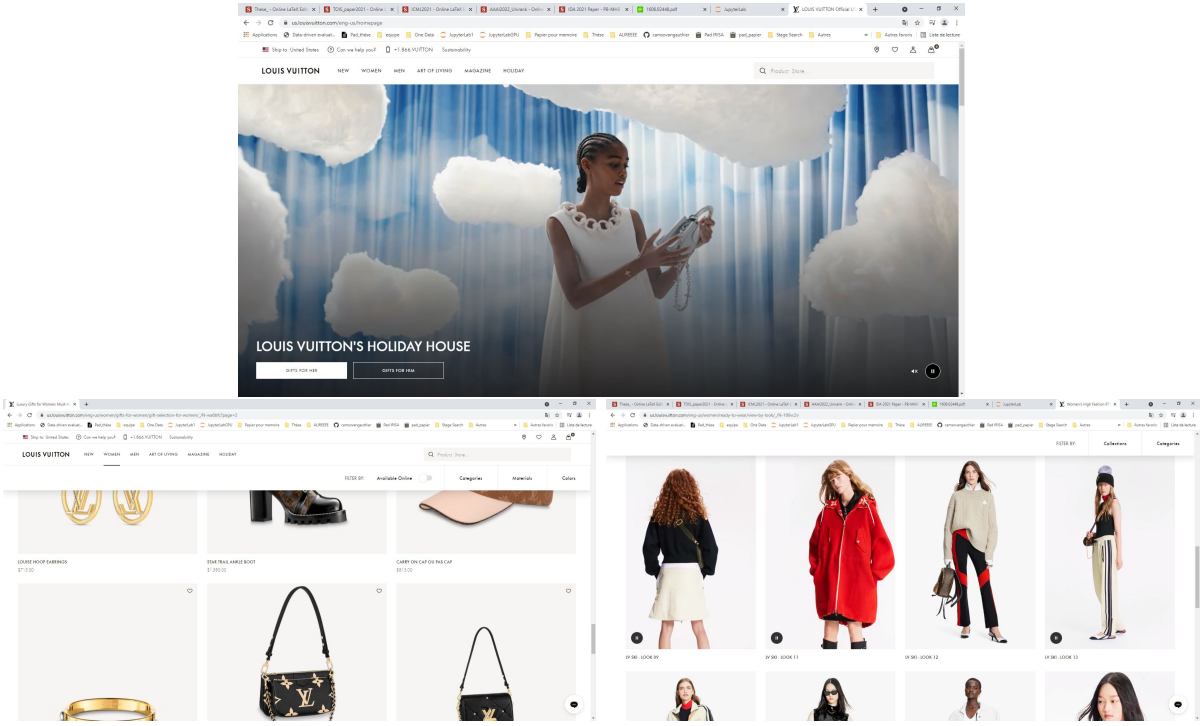


Figure 4: Pages of the Louis Vuitton website. Top: Homepage. Bottom left: "Women Gift" section with a grid display. Bottom right: "Ready to Wear by Look" section, with multiple product displayed on looks on a grid.

world and those which do not. To increase its accessibility, Louis Vuitton has various channels: its website, its chatbot, called the Virtual Assistant, its application "LV app" and social and mobile applications such as Instagram, Wechat or Line. On each channel, dedicated teams are in charge of designing *experiences* to satisfy their clients' needs. For instance, Figure 4 presents three different experiences on the website: the homepage dedicated to highlight LV news, a grid of product dedicated to showcase gift suggestions and a grid of "looks" to present ready-to-wear products. The Virtual Assistant is also able to answer in an engaging and interactive way with clients. For instance, Figure 5 shows a Product Discovery query from a user.

Nevertheless, clients have high expectations towards the experience proposed by Louis Vuitton and, because the share of distant sales is increasing, these expectations have influenced the design and management of all the communication channels. Currently, on Louis Vuitton's website, recommendations are based on business rules, visual similarity between products or products' popularity and past actions of clients to address clients' needs for generic advises, alternative recommendations and to push contents. Since there

is frequently new products included in LV products' catalogue, recommendations on the website also need to continuously include these new products.

The use cases presented in this section have in common to choose and rank K items among L possible ones. This setting is known as list recommendation and will be the focus of this thesis.

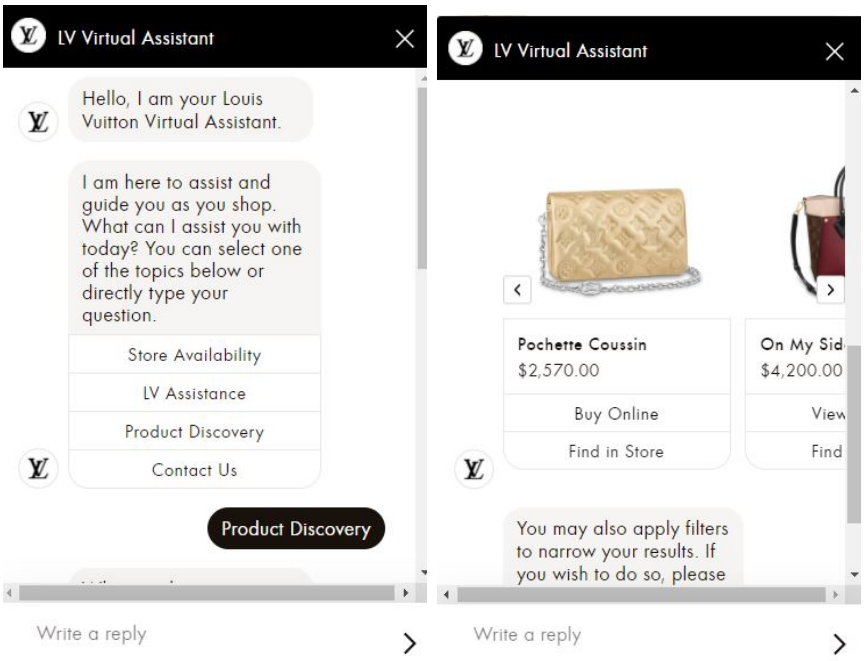


Figure 5: Product Discovery on Louis Vuitton's Virtual Assistant.

Recommendation systems in our everyday life

Recommendation Systems (RS) are designed to assist users to choose relevant items. These items can be songs, adds or movies and are often meant to be displayed on web pages, among a very large number of elements. When a human sales advisor would have walked you through all the shelves of a physical store, a Recommendation System can help thousands of web users to search in millions of products at the same time. This automatic process helps companies to satisfy their digital clients. RS cover many needs in terms of interactions with clients: completion of basket, alternative choices, ... they can focus on a single recommendation or provide multiple ones at a time to give more freedom to users. At each call, such systems select K items among L potential ones, $K \leq L$. User feedbacks are collected for each displayed items, reflecting how relevant these automated choices are:

listening time, clicks, rates, etc. Usually, these feedbacks are used through batch learning by applying for instance collaborative filtering methods [57], content-based ones [47] or through embedding alongside features describing users and items [10, 65]. However, these feedbacks are available only for the items which were actually presented to the user. *Online Recommendation Systems* (ORS) are developed to overcome this issue. The *Multi-armed bandit problem* with *semi-bandit feedback* [19, 9] is a standard way to describe this setting. This setting tends to iteratively recommend one or multiple items amongst a larger set of possible items, each of them being independent from the other, then it receives a feedback for each item recommended which will be used by the algorithm to choose the next list of recommendations. This use of feedbacks drives the algorithm to adopt two types of behavior: on the one side, the algorithm presents items with few or no previous feedback at to users to collect information on all the possible items and on the other side, the algorithm promotes items which have the best feedbacks.

Another problem, related to ranking, is to display the K chosen items at the right positions to maximize the user attention. Typical examples of such displays are (i) a list of news, visible one by one by scrolling; (ii) a list of products, arranged by rows; or (iii) advertisements spread everywhere on a web page. Numerous approaches have been proposed to jointly learn how to choose the best positions for the corresponding best items [53, 15, 44] referred to as *multiple-play bandit* or *online learning to rank* (OLR).

Online learning to rank at Louis Vuitton

To tackle the online learning to rank problem arising from LV's use cases, one needs to understand and identify the click behavioral model followed by LV users. Then, algorithms can be developed to efficiently infer the parameters of the relevant model.

On Louis Vuitton's website, for example, products are displayed on grids of different shape, depending on your device (phone, computer...). These different types of display lead to various reading directions for users. Thus, it is important to understand how clients interact with the displayed recommendations as clients' attention toward a displayed item is impacted by its position on the grid. By putting the selected products in positions which will be seen by clients, their experience will be enhanced and they will find more easily what they need.

The client gives some feedback when presented with a recommended list (i.e. a ranked list of K items). We consider here that a feedback is a list of K Booleans (click or no click),

one for each presented item. Behavioral click models aim at providing a parametric model of the interactions between the clients and the recommended lists. This model defines the click probabilities for each item given a recommended list. Many click behavioral models are identified to understand how clients provide their (partial) attention [55, 16]. The main issue is to understand if an item located at a given position has been seen or not and thus if an item is not receiving a click because it was not relevant or because it has not been seen. This probability of click should depend both on the relevancy of an item and the impact of its position. Different existing click behavioral models implement this assumption differently: the position-based model (PBM) assumes that the relevancy of an item and the impact of its position are independent; The cascading model (CM) assumes that clients look at positions from top to bottom.

Once the most relevant click behavioral model is identified, parameters have to be infered through an efficient algorithm. LV's website attracts many visitors each day, and showcases the product catalogue of LV, which contains tens of thousands of items and is renewed frequently. Moreover, the website display is currently mostly static with some changes when new business rules arrive. These facts combined, lead to very few interactions per product per client and these interactions are focused on the same products. This is harmful for traditional recommendation approaches such as collaborative filtering and matrix factorisation. To overcome this concentration on few information, bandit algorithms can be implemented as complementary module to existing recommendation systems. Through their continuous learning, bandit algorithms make the display more dynamic and learn on all products thanks to both users interactions and bandits' exploration component. As we are displaying K items simultaneously and collecting the interactions on each item, we face the, so called, multiple-play semi-bandit setting [9].

Summary of the thesis goals

To tackle the particular recommendation use cases encountered at Louis Vuitton, we focus on the *online learning to rank* problem. To apply online learning to rank in these cases, we first need to identify the right click behavioral models for LV's clients and develop new bandit algorithms to efficiently infer the parameters of such click behavioral models.

Thesis outline

This thesis is organised as follows. Chapter 1 presents general ideas on the two main concepts used in this thesis: behavioral click models and bandit algorithms. Various click models are depicted (Section 1.1) both in terms of user interaction but also according to general assumptions on the definition of the probability of clicks. A general framework for bandit algorithms and some details on a few algorithms such as Thompson sampling and Upper Confidence Bound (UCB) are presented in Section 1.2. Section 1.2.4 narrows the focus to combinatorial bandits, which have similar issues as some of our contributions, more precisely "how to allocate products to positions" can be seen as making a combinatorial recommendation. To conclude this chapter, section 1.3 presents the setting used in this thesis. A focus on the evaluation is made by defining the performance metrics, the feedbacks generation process and the benchmark datasets.

Chapter 2 provides algorithms related to the setting presented in the previous chapter and to the contributions presented in the subsequent chapters. The algorithms are divided into three groups: algorithms specialized in the position-based model (PBM), algorithms designed for other click models and for unspecified click models and finally, algorithms which structure is close to the contributions presented in this thesis.

Chapter 3 presents a contribution based on unimodal bandits and published in [26] and presented at [25]. This adaptation of the unimodal bandit framework to our position-based setting is presented in Section 3.1. Then, our algorithm, GRAB (for parametric Graph for unimodal RAnking Bandit), is presented in Section 3.2. Theoretical analysis and practical results are presented in Sections 3.3 and 3.4.

Chapter 4 presents a contribution to tackle specifically PBM by coupling Thompson Sampling bandits and Markov chain Monte Carlo approximation sampling methods. This work has been published in [23] and presented at [24]. An extended version of [23] is under review in *ACM Transactions on Information Systems*. Section 4.1 presents the two approximation methods used in this contribution with the associated algorithms: PB-MHB (for Position Based Metropolis-Hastings Bandit) and PB-LB (for Position Based Langevin gradient Bandit). Section 4.2 gives empirical results and shows that, even though this contribution is less well theoretically grounded than the one presented in the previous chapter, it provides the best performing algorithms.

Chapter 5 presents a contribution designed to recommend under a wider setting than in the two previous chapters. This contribution, called UniRank (for Unimodal Bandit

Algorithm for Generic Online Ranking) adapts unimodal bandits to tackle multiple click models. These adaptations are presented in Sections 5.1 and 5.2. Theoretical analysis and practical results are given in Sections 5.3 and 5.4.

Finally, we conclude by giving the main take away of these three contributions and by discussing the social impact of such systems. Some perspectives of this work both from a research point of view and from an industrial one are then given.

Publications

Here is the list of the articles written in the context of this thesis:

- Published in international venues [23] and [26]:
  - *Camille-Sovanneary Gauthier, Romaric Gaudel, Elisa Fromont*, "Bandit Algorithm for both Unknown Best Position and Best Item Display on Web Pages", IDA: 19th International Symposium on Intelligent Data Analysis, Porto,Portugal, 2021.
  - *Camille-Sovanneary Gauthier, Romaric Gaudel, Elisa Fromont and Aser Boammani Lompo*, "Parametric Graph for Unimodal Ranking Bandit", ICML: Proceedings of the 38th International Conference on Machine Learning, virtual, 2021.
- Published in national venues [24] and [25]:
  - *Camille-Sovanneary Gauthier, Romaric Gaudel, Elisa Fromont*, "Bandits manchots avec échantillonnage de Thompson pour des recommandations basées sur les positions", CAp'2020 (Conférence d'Apprentissage), virtual, 2020.
  - *Camille-Sovanneary Gauthier, Romaric Gaudel, Elisa Fromont and Aser Boammani Lompo*, "Ordonnancement d'objets par bandits unimodaux sur des graphes paramétriques", CAp'2021 (Conférence d'Apprentissage), Saint-Etienne, 2021.
- Under review (journal):
  - *Camille-Sovanneary Gauthier, Romaric Gaudel, Elisa Fromont*, "MCMC-based Thomson Sampling Algorithms for Online Recommendations in the Position-Based Model", ACM Transactions on Information Systems (TOIS).
- To be submitted at ICML 2022 (rejected at NeurIPS 2021 and improved):
  - *Camille-Sovanneary Gauthier, Romaric Gaudel, Elisa Fromont*, "UniRank: Unimodal Bandit Algorithm for Generic Online Ranking".

Chapter 1

Background on bandit-based recommender systems

Contents

1.1 User click behavioral models 28
1.1.1 Position-based model 29
1.1.2 Cascading model 30
1.1.3 Others click behavioral models 31
1.2 Bandit algorithms 31
1.2.1 Generality 32
1.2.2 Thompson sampling 33
1.2.3 Upper confidence bound algorithm 34
1.2.4 Combinatorial bandits 35
1.3 My thesis setting: learning to rank in a semi-bandit setting . 36
1.3.1 Bandits for click behavioral model 36
1.3.2 Performance evaluation 37
1.3.3 Choice of the environment to evaluate bandit algorithms . . . 38
1.3.4 Datasets 39
1.4 Conclusion 42

This chapter provides general ideas about the main scientific concepts tackled in this thesis: click behavioral model and bandit algorithms.

Behavioral clicks models provide a framework for users' interactions with web pages and how to interpret these interactions. In particular, they give some information about *which* knowledge can be extracted from such interactions and *which* assumptions should be considered.

Then, these interactions can be used by bandit algorithms in various ways to learn user preferences. Some of these general methods are described here to understand *how*

this information is used in order to optimize recommendations and adapt to users.

Finally, these general presentations lead to the introduction of our setting. We describe *which* paths we explore to address the general problematic depicted in the Introduction, and *which* choices we made to formalize our problem and test our ideas.

1.1 User click behavioral models

Everyone has its own attention. However when you want to provide a piece of information to someone, you have to understand his/her attention mechanism in order to make sure that your information is received.

When the information goes through a digital interface, identifying users' attention is harder as digital browsing does not leave any clear signal. The only signal one has access to is the user feedback, often his/her clicks on a piece of information displayed. This piece of information will be referred to as an *item*. Clicks on items mix the attention of the user with the relevance of the item which has been clicked on. In other words, when an item is not clicked on, one does not know if it is because the user did not see it (the item did not get the user attention) or because the user did not like the item (the item was irrelevant). Nevertheless, as attention is mostly linked to how the information is transmitted, for instance how items are placed on a web page, many models were designed to understand users' click behavior toward the proposed display.

In our work, we focus on a setting where multiple items are displayed at the same time. In the following, we always denote K the number of items we want to display at a time, chosen from a set of L available ones ($L > K$). We represent the K displayed items as the list $\boldsymbol{a} = (a_1, \dots, a_K)$, where for each $k \in [1, K]$, a_k is the item displayed at the k-th position. We also define $\mathcal{A}$ the set of all possible lists of K distinct elements among L. Feedbacks of a user according to $\boldsymbol{a}$ are noted $\boldsymbol{c} = (c_1, \dots, c_K)$ and are associated here with the fact that the user clicked or not on an item. Thus, for $k \in [1, K], c_k \in \{0, 1\}$ is the fact that a user clicks or not on the k-th item when $\boldsymbol{a}$ is displayed.

Usually, the marginal probability of interaction towards the item placed at the k-th position is decomposed in two terms:

$$\mathbb{P}(c_k|\boldsymbol{a}) \overset{def}{=} \chi(\boldsymbol{a}, k)\theta_{a_k}, \tag{1.1}$$

where $\chi(\boldsymbol{a}, k)$ is interpreted as the probability that the user looks at the position k given $\boldsymbol{a}$, and θ_{a_k} represents the probability for the user to click on the item a_k when the user looks at it. Note that this decomposition does not assume anything regarding the link between clicks c_k and $c_{k'}$ at two different positions $k \neq k'$; these clicks may either be correlated or independent. Similarly, we do not make any assumption on the way this recommendation is displayed, for example it can take the shape of a list, a grid or it can be spread on a (web) page.

This general definition can be instantiated into more specific click behavioral models such as the widely studied Position-based model (PBM) and Cascading model (CM) which are described respectively in Sections 1.1.1 and 1.1.2. These models are defined by referring to the way users interact with $\boldsymbol{a}$, but we can also define axioms on $\chi(\boldsymbol{a}, k)$ and θ_{a_k} to encompass a wider range of behaviors, as it is shown in Section 1.1.3.

1.1.1 Position-based model

The Position-Based Model (PBM) [55, 16] relies on two vectors of parameters: $\boldsymbol{\theta} \in [0,1]^L$

and $\boldsymbol{\kappa} \in [0,1]^K$, where θ_i is the probability for the user to click on item i when he/she observes that item, and κ_k is the probability for the user to observe the position k. These parameters are unknown, but they may be inferred from user behavior data: we need to first record the user feedback (click vs. no-click per position) for each set of displayed items, then we may apply an *expectation-maximization* framework to compute the maximum a posteriori values for $(\boldsymbol{\theta}, \boldsymbol{\kappa})$ given these data [12].

More formally, for $\boldsymbol{a} \in \mathcal{A}$ and $k \in [1, K]$, this model defines the feedback associated with each couple (item, position) (a_k, k) as the product of two independent random variables: X_{a_k}, which is the result of the event "the user finds a_k relevant" and Y_k, which is the result of the event "the user sees the position k". We have:

$$\begin{aligned} X_{a_k} &\sim Ber(\theta_{a_k}), \\ Y_k &\sim Ber(\kappa_k), \\ c_k &= Y_k X_{a_k}, \end{aligned}$$

where Ber is the Bernoulli distribution. As every variable are iid, we can write:

$$c_k \mid \boldsymbol{a} \overset{iid.}{\sim} \text{Ber}\left(\theta_{a_k}\kappa_k\right), \tag{1.2}$$

in other word,

$$\begin{cases} \mathbb{P}\left(c_k = 1 \mid \boldsymbol{a}\right) = \theta_{a_k}\kappa_k, \\ \mathbb{P}\left(c_k = 0 \mid \boldsymbol{a}\right) = 1 - \theta_{a_k}\kappa_k. \end{cases}$$

The general form expressed by Equation (1.1) can be found in this definition of PBM by taking $\chi(\boldsymbol{a}, k) = \kappa_k \ \forall k \in [1, K]$. PBM is particularly interesting when the display is dynamic, as often on modern web pages, and may depend on the reading direction of the user (which varies from one country to another) and on the ever-changing layout of the page.

1.1.2 Cascading model

The cascading model [16] is another popular user click behavioral model. It assumes that the positions are observed in a known order and that the user leaves the website as soon as he/she clicks on an item. More specifically, if the user clicks on the item in position k, he/she will not look at the following $k+1, \ldots, K$ positions.

To define c_k, L independent random variables X_{a_k} are drawn from a Bernoulli distribution $Ber(\theta_{a_k})$ and the click on each element of $\boldsymbol{a}$ is defined by:

$$\begin{aligned} X_{a_k} &\sim Ber(\theta_{a_k}), && \forall k \in [1, K], \\ c_k &= X_k \prod_{j=1}^{k-1}(1 - X_j), && \forall k \in [1, K]. \end{aligned}$$

While in PBM, a user can click on more than one element of $\boldsymbol{a}$, in CM, user can click on at most one element. The user will either click on one element and leave or he/she will see all the elements of $\boldsymbol{a}$ and leave without clicking.

Here, again, the general form meets this definition of CM with $\chi(\boldsymbol{a}, k) = \prod_{j=1}^{k-1}(1 - \theta_{a_j})$ $\forall k \in [1, K]$. CM is commonly used to describe users' click behavior toward a list of propositions such as the output of standard search engines.

1.1.3 Others click behavioral models

There exist many more click behavioral models which fall in the general setting defined by Equation 1.1. For instance, the Dependant Click Model (DCM) which can be seen as an extension of CM with the addition of a probability of leaving which reflects the "patience" of the user to leave the consultation of $\boldsymbol{a}$ before seeing everything and without clicking on any seen item. All the models described previously adapt the Equation 1.1. This model descriptions interprets Equation 1.1's terms as clients interaction. Here, $\chi(\boldsymbol{a}, k)$ reflects users' interactions and gives its specificity for both PBM and CM. For instance in CM, $\chi(\boldsymbol{a}, k)$ is the probability that no element was clicked on before reaching position k.

Another way to describe a user behavior is to directly put assumptions on $\chi(\boldsymbol{a}, k)$ and θ_{a_k} as done, for example, in [43]. We give, here, examples of such assumptions:

— Let $\boldsymbol{a}^* \in \mathcal{A}$ be an optimal proposition [1]. Then $\max_{a \in \mathcal{A}} \sum_{k=1}^{K} \mathbb{P}(c_k|\boldsymbol{a}) = \sum_{k=1}^{K} \mathbb{P}(c_k|\boldsymbol{a}^*)$. This assumption states that the optimal action orders the items according to the order in which each positions are seen.

— Let i and j be items with $\theta_i \geq \theta_j$ and let $\sigma_{i,j} : \mathcal{A} \to \mathcal{A}$ be the permutation that exchange i and j and leaves other items unchanged. Then for any $\boldsymbol{a}$, where i is the position k, $\mathbb{P}(c_k|\boldsymbol{a}) \geq \frac{\theta_i}{\theta_j} \mathbb{P}(c_k|\sigma_{i,j}(\boldsymbol{a}))$, with $\sigma_{i,j}(\boldsymbol{a})$ be the proposition resulting from applying the permutation σ to $\boldsymbol{a}$. This assumption expresses the impact of reversing two elements which were ordered correctly.

These assumptions include all three models described before, and more, without making assumptions on the type of interactions between the user and the proposed list of items.

1.2 Bandit algorithms

To cope with the real world where perfect-information does not exist, we may need to make repeated choices under uncertainty.

In order to make good choices ultimately, what strategy should a learner adopt? Should he/she focus on exploration and then choose according to the data collected? How many choices should focus on exploration? How to use information collected on the fly to benefit quickly and reasonably from this collect? This last point is especially relevant in the digital

1. optimality is defined in Section 1.2, but as it is implied the optimal proposition is best possible list to display to the user

realm where it is easier to proceed by test and learn. It is also the place of everlasting changes thus being dynamic is a compulsory feature.

Moreover, when facing recommendations, a user has only access to the items proposed to him/her. How to collect information, while optimizing without spoiling the collection by being too greedy?

Bandit algorithms are designed to face this dilemma called the *exploration/exploitation dilemma*. They are often used in a complementary way to more "traditional" recommendation systems to specifically enhance the way information are collected.

1.2.1 Generality

The Bandit framework consists in making $T \in \mathbb{N}$ consecutive interactions, with T called the *horizon*. At each iterations $t \in T$, an interaction stands between a learner, who takes an action a_t from a set of actions $\mathcal{A}$, and an environment ν, which gives a feedback, a *reward* r_t, to the learner according to a_t.

An environment ν is the set of laws over each action. Drawing from ν at iteration t leads to the vector $\boldsymbol{x}(t) = (x_1(t), \ldots, x_{|\mathcal{A}|}(t)) \sim \nu$ which couples each possible action to its reward. At each iteration the reward r_t retrieved from taking action a_t is $r_t = x_{a_t}(t)$.

The learner bases its iterative choices on the actions and rewards gathered at previous iterations. It is noted $D_t = \{(a_1, r_1), \ldots, (a_{t-1}, r_{t-1})\}$. The use of D_t to take action is called a policy, π. We have $\pi(D_t) = a_t$. Each policy iteratively build D_t from $D_0 = \{\}$ by adding (a_t, r_t) at each iteration t. Note that even if the notation related to π reminds a function, π is a way of choosing the next action to take, given the information available. The main goal of the learner is to choose a policy which maximises the cumulative reward ($\sum_{t=1}^{T} r_t$) over all T steps. To do so, the learner does not have access to the actual environment it is interacting with but, to a set of environment $\mathcal{E}$, it has to choose from. For instance, $\mathcal{E}$ can be a set of models with unknown fixed parameters, the learner will have to infer the parameters fitting the data collected, in order to apply an appropriate policy. In iterative choices, bandits offer a great addition to more traditional information retrieval methods as they can take into account the dependency of previous choices to the next one.

We note a^* the best choice over the T iterations, a.k.a. the optimal action. This optimal action is assumed to be the same over all T iterations. a^* maximises the cumulative expected reward, $\mu_{\boldsymbol{a}} = \sum_{t=1}^{T} \mathbb{E}[r_t | a_t = \boldsymbol{a}]$. The policy which leads to a^*, called optimal policy is noted π^* and for each iteration t, $\pi^*(D_t) = a^*$. Thus, we have $\mu^* = \sum_{t=1}^{T} \mathbb{E}_{\pi^*}[r_t] = \sum_{t=1}^{T} \mathbb{E}[r_t | a_t = \pi^*(D_t)]$.

Algorithm 1 Thompson Sampling algorithm

Require: prior law $\mathcal{U}$
Require: posterior law Q
  $D_0 \leftarrow \{\}$
  $\nu_0 \sim \mathcal{U}$
  for $t = 1, 2, \ldots, T$ **do**
    choose $a_t = argmax_{a \in \mathcal{A}} \mu_a(\nu_{t-1})$
    observe reward r_t
    $D_t \leftarrow (a_t, r_t)$
    $\nu_t \sim Q(D_t)$
  end for

In order to evaluate a learner, the usual measure is the *cumulative expected regret* R_T rather than the cumulative expected reward. R_T denotes the loss of a learned policy compared to the optimal policy.

$$R_T \stackrel{def}{=} \sum_{t=1}^{T} \mathbb{E}_{\pi^*}[r_t] - \sum_{t=1}^{T} \mathbb{E}_{\pi}[r_t]. \tag{1.3}$$

Among all the possible policies, two major families are presented in Sections 1.2.2 and 1.2.3. The last section of this chapter gives some details about a specific bandit setting: *combinatorial bandits*.

1.2.2 Thompson sampling

The Thompson Sampling (TS) algorithm [62] is the first bandit policy ever described. It uses Bayesian inferences to explore. It adds noises to the parameters estimators and allows to control the randomized choice among actions.

The TS algorithm (see Algorithm 1) starts with an environment $\nu_0 \in \mathcal{E}$ drawn from a prior distribution $\mathcal{U}$ over $\mathcal{E}$. Then, at each iteration t, a new environment ν_t is drawn from a posterior distribution Q which maps D_t over $\mathcal{E}$. Then, the action is chosen by maximizing the expected gain in this environment. For the ith action in ν_t, this expected gain is defined as $\mu_i(\nu_t) = \int_{\mathbb{R}} x_i dP_{\nu_t}(\boldsymbol{x})$. Q is the reward distribution and U is either neutral or is constructed on a practical belief over the actions.

Thompson sampling policy is known to have good empirical performances on a wide range of environments. The main limitation is the restrictions over Q. It has to be easy

Algorithm 2 Upper Confidence Bound algorithm

Require: $\mathcal{A}$
  for $t = 1, \ldots, |\mathcal{A}|$ **do**
    $D_0 \leftarrow \{\}$
    Choose $a_t = t$
    observe reward r_t
    $D_t \leftarrow (a_t, r_t)$
  end for
  for $t = |\mathcal{A}|, \ldots, T$ **do**
    choose $a_t = argmax_{a \in \mathcal{A}} UCB_a$
    observe reward r_t
    $D_t \leftarrow (a_t, r_t)$
  end for

to draw from in order to proceed to the inference step. If the distribution does not have a closed form, an approximation of the draw is needed.

1.2.3 Upper confidence bound algorithm

Upper Confidence Bound algorithms [4] are bandit algorithms which base their trade-off between exploration and exploitation on the principle of *optimism in face of uncertainty*. This principle promotes exploration over actions which have been chosen too few times and adjusts the probability of choosing an action according to the upper confidence bound on the expected reward. These algorithms choose the next action by taking the $argmax$ of this bound (see Algorithm 2).

There exist many variations of this method to take into account the specificities of the encountered setting. The most common one is UCB1. This algorithm is designed to tackle a setting were there is a finite number of possible actions. These actions are independent and their rewards are bounded. UCB1 defines its bound as:

$$UCB_a^{UCB1} = \hat{\mu}_a + \sqrt{\frac{2\log(t)}{T_a}},$$

where $\hat{\mu}_a$ is an estimation of the expected cumulative reward's value for a and $T_a = \sum_{t=1}^{T} \mathbb{1}_{a_t=a}$ is the number of times a has been chosen so far.

An alternative to this algorithm is based on the *Kullback-Leibler divergence* [21]. It

leads to the following bound:

$$UCB_a^{KL} \stackrel{def}{=} f\left(\hat{\mu}_a, T_a, t\right), \tag{1.4}$$

where $f(\hat{\rho}, s, t)$ stands for

$$\sup\{p \in [\hat{\rho}, 1] : s \times \mathrm{kl}(\hat{\rho}, p) \leq \log(t) + 3\log(\log(t))\},$$

with

$$\mathrm{kl}(p, q) \stackrel{def}{=} p \log\left(\frac{p}{q}\right) + (1-p)\log\left(\frac{1-p}{1-q}\right),$$

the *Kullback-Leibler divergence* from a Bernoulli distribution of mean p to a Bernoulli distribution of mean q; $f(\hat{\rho}, s, t) \stackrel{def}{=} 1$ when $\hat{\rho} = 1$, $s = 0$, or $t = 0$.

This definition of the UCB^{KL} achieves better performances when the setting is the same as UCB1, namely a limited number of independent actions, with bounded rewards.

1.2.4 Combinatorial bandits

After seeing two policies common in the bandit literature, the previous definition of an environment is extended here to reach the setting used in this thesis and detailed in the last section of this chapter.

Stochastic combinatorial semi-bandits [19, 9] are online learning problems where a learner has to choose a subset of items under some given combinatorial constraints. Then the learner observes some stochastic weights for each item and receives as a payoff, the result of a function of these weights. This function depends on the combinatorial problem the bandit is facing. In the following section, this function is considered as the *sum* of these weights. This combinatorial setting is a way to split the initially more complex arms into simpler elements, for instance a list of products can be seen as a complex arm composed of each product beeing its simpler elements. Then, in combinatorial setting, the algorithm learns the impact of each element. In many cases, observations made on one arm benefit others. By splitting these observations to each element, a shared element brings knowledge to the other arms which use this element. [2] This setting prevents an exponential increase in the number of arms, called the combinatorial explosions. It is also possible to add constraints on these associations of elements.

2. In some cases, the bandits have only access to the combination of the weights of the chosen arm's elements. This situation is referred to as *stochastic combinatorial bandits*.

Combinatorial learning problem can be written as a tuple $B = (E, \mathcal{A}, \nu)$. E is a set of elements, $\mathcal{A} \subseteq \{0,1\}^{|E|}$ is a set of arms, where each arm $\boldsymbol{a}$ is a subset of E and ν is a distribution on weights. Each weight translates conditions to combine the element. For example, in a case where you have to propose a subset of K element amongst L, each element wil have a weightFollowing the terminology used in [40], E is called the *ground set* and $\mathcal{A}$ the *feasible set*.

At each iteration, the bandit algorithm chooses a subset of elements $\boldsymbol{a} \in \mathcal{A}$ and receives the reward $\sum_{e \in \boldsymbol{a}} w_e$, where $\boldsymbol{w}$ is an independent draw of ν on $[0,1]^{|E|}$. The expected reward associated to the element e is denoted $\rho_e \overset{def}{=} \mathbb{E}_{\boldsymbol{w} \sim \nu}[w_e]$. As choosing $\boldsymbol{a} \in \mathcal{A}$ is equivalent to choose each element of $\boldsymbol{a}$, the expected reward when choosing the arm $\boldsymbol{a}$ is:

$$\mu_{\boldsymbol{a}} \overset{def}{=} \mathbb{E}_{\boldsymbol{w} \sim \nu}\left[\sum_{e \in \boldsymbol{a}} w_e\right] = \sum_{e \in \boldsymbol{a}} \rho_e. \tag{1.5}$$

The best expected reward is, as before, noted $\mu^* \overset{def}{=} \max_{\boldsymbol{a} \in \mathcal{A}} \mu_{\boldsymbol{a}}$.

The learning agent interacts with the environment T times and its goal is to maximize $\mu_{\boldsymbol{a}}$ over the T steps. If the learner knew ν a priori, the optimal action would be to choose $a^* = \text{argmax}_{a \in \mathcal{A}} \mu_{\boldsymbol{a}}$ at all steps t.

As for classical bandit settings, methods such as UCB can be used to estimate each ρ_e and thus $\mu_{\boldsymbol{a}}$. One of the main challenge of these settings is the definition of the combinatorial constraints which define $\mathcal{A}$ and the possible arms to recommend.

1.3 My thesis setting: learning to rank in a semi-bandit setting

As already mentioned in the introduction, we are facing a recommendation problem where one has to recommend K items among L possible ones. These items are displayed on a grid.

1.3.1 Bandits for click behavioral model

As we recommend a (ordered) list of items, we consider the following *online learning to rank (OLR) problem with clicks feedback*. At each iteration t, a recommendation $\boldsymbol{a} = (a_1, \ldots, a_K)$ is chosen among $\mathcal{A}$ and displayed to the user. In this setting, $\mathcal{A}$ is the set of all possible ordered lists of K items among the L possible ones. We can define $\mathcal{A} = \mathcal{P}_K^L$, the set

of permutations of K items among L as a permutation is the ordered choice of K distinct items among L. Let a_k be the item displayed at position k. $\boldsymbol{a} \stackrel{def}{=} \{a_k : k \in [K]\}$ is the set of all displayed items. Throughout the thesis, the terms *permutation* and *recommendation* are used interchangeably to denote an element of $\mathcal{P}_K^L$.

An instance of our OLR problem is a tuple (L, K, ν), where L is the number of available items, $K \leqslant L$ is the number of positions to display the items, and ν is a set of distributions from $\mathcal{A} \times [L]$ to $[0, 1]$ such that for any recommendation $\boldsymbol{a}$ and position k, $\nu(\boldsymbol{a}, k)$ is the distribution of probability that a user clicks on the item displayed at position k when recommending $\boldsymbol{a}$. As we are transposing click behavioral models to the bandit framework, we have $\mathbb{E}[\nu(\boldsymbol{a}, k)] = \mathbb{P}(c_k|\boldsymbol{a})$ with $\mathbb{P}(c_k|\boldsymbol{a})$ defined according to the chosen model.

In our setting, a recommendation algorithm is only aware of L and K and has to deliver T consecutive recommendations. At each iteration $t \in [T]$, the algorithm recommends a permutation $\boldsymbol{a}(t)$ and observes the reward $r_t = \boldsymbol{c}(t) = (c_{a_1(t)}(t), \ldots, c_{a_K(t)}(t))$, where for any position k, $c_{a_k(t)}(t)$ equals 1 if the user clicks on the item $a_k(t)$, and 0 otherwise. To keep notations simple, we also define $c_i(t) = 0$ for undisplayed items $i \in [K] \setminus \boldsymbol{a}(t)$. Recall that the recommendation at time t is only based on previous recommendations and observations.

In this thesis, we tackle a special case of this general OLR, named PB-OLR for position-based online learning to rank. An instance of a PB-OLR problem is a tuple $(L, K, (\rho_{i,k})_{(i,k)\in[L]\times[K]})$. For any item i and position k, $\rho_{i,k}$ is the probability for a user to click on item i when displayed at position k, independently of the items displayed at other positions. It is a particular case where ν is adapted to settings where the position's impact must be taken into account. Under the PBM click model, there exist two vectors $\boldsymbol{\theta} \in \mathbb{R}^L$ and $\boldsymbol{\kappa} \in \mathbb{R}^K$, such that $\rho_{i,k} = \theta_i \kappa_k$ (i.e. $\boldsymbol{\rho}$ is of rank 1). It is a subcase of this broader PB-OLR problem.

Overall, we are facing a Bernoulli semi-bandit setting as we are facing an online learning to rank problem with access to a vector r_t of click feedbacks. Depending on the click behavioral model assumed, we can defined our performance measure.

1.3.2 Performance evaluation

Section 1.2.1 gave a general definition of the cumulative expected regret R_T as a standard evaluation metric. We detail here its definition, when facing our semi-bandit setting.

In this setting, as the individual clicks are observed, the reward of the algorithm is

their sum $r_t \overset{def}{=} \sum_{k=1}^{K} c_{a_k(t)}(t)$, while recommending $\boldsymbol{a}(t)$. Let $\mu_{\boldsymbol{a}}$ denote the expectation of r_t when the recommendation is $\boldsymbol{a}(t) = \boldsymbol{a}$, and $\mu^* \overset{def}{=} \max_{\boldsymbol{a} \in \mathcal{P}_K^L} \mu_{\boldsymbol{a}}$ the highest expected reward. Once again, the aim of the algorithm is to minimize the expected cumulative regret

$$R_T = T\mu^* - \mathbb{E}\left[\sum_{t=1}^{T} \mu_{\boldsymbol{a}(t)}\right], \tag{1.6}$$

where the expectation is taken w.r.t. the recommendations from the algorithm and the clicks. This definition can adapt to all the presented behavioral settings by expressing μ according to the expected reward of these models.

For instance, applying this definition in PBM leads to $\mu_a = \sum_{k=1}^{K} \theta_{a_k} \kappa_k$ and $R_T = \mu^* T - \sum_{t=1}^{T} \sum_{k=1}^{K} \theta_{a_k(t)} \kappa_k$.

Remark 1 (Regret's limitation)**.** The expected cumulative regret is the standard measure to evaluate the performance of bandit algorithms, but it requires the knowledge of the optimal recommendation. To control this knowledge and show the specific behavior of a method, an offline evaluation on simulated data is often done in practice. Nevertheless, it may lead to unrealistic situations. Moreover, in "business" practice, this optimal recommendation is hard to get, due to changes in customer preferences or to the addition of new items. An alternative measure is the *cumulative reward*. Both measures lead to the same conclusions, but the cumulative expected regret gives additional insight on the impact of the learning strategy. Thus this regret evaluation measure will be preferred.

1.3.3 Choice of the environment to evaluate bandit algorithms

As mentioned in Remark 1, regret is a more precise measure than reward. It can be applied in various situations. In this section, we elaborate on the design of an appropriate environment to compute the regret and evaluate our algorithms while staying close to real life situations.

To evaluate the impact of iterative recommendation systems, online solution can be considered such as deploying the algorithm in A/B test situations. Nevertheless, it requires a careful designed of the A/B test and firms are more inclined to have offline evaluations before putting a system into production. So want to measure the gain of a change before applying it.

Thus, the experiences performed throughout this thesis are done offline and measured

using the regret R_T. To reconcile offline and realistic data, many experience settings exist. Offline experiences on bandit algorithms are based on log datasets recording recommendations and users' feedbacks. They allow to measure regret at each iteration.

Some evaluations are, for instance, based on replay methods [42]. Applied strictly, at an iteration t, a replay method uses a log data and when the logged action mismatches the action chosen by the evaluated policy, the iteration is discarded from the evaluation. It decreases the number of records used by $1/|\mathcal{A}|$, with $\mathcal{A}$ the set of possible arms.

Many evaluation methods are built upon the replay idea such as bootstrap sampling replay [51] or counterfactual estimators on list recommendations [60]. These methods use the distribution on the arm induced by the policy which has collected the data to evaluate on more logged data, and thus a wider time horizon T. Another usual way to get around this data dropping issue is to simulate the interactions according to the real-life logged data. To do so, the parameters of the assumed click behavioral model are inferred and then used to simulate new interactions. For instance, let us imagine that an online recommender system is required to deliver T consecutive recommendations. At each iteration t, the user feedback is drawn from a distribution derived from the PBM distribution given in Equation (1.2), with $\boldsymbol{\theta}$ and $\boldsymbol{\kappa}$ inferred from the logged data.

Since this method is often chosen in our closest related work, it is also our chosen evaluation strategy in this thesis for all our algorithmic contributions. The inference of the parameters used to simulate the click behavior is detailed in the next section, along with a description of the datasets that are used in this thesis.

1.3.4 Datasets

Two types of data are considered in our experiments. They are denoted *purely simulated*, for simulated data and *behavioral*, for data based on real life datasets. For both of them, the feedbacks are simulated by drawing from a set of distributions ν. The parameters of the distributions in ν are chosen to reflect specific situations in purely simulated experiments. The values of the parameters of the tested click behavioral models are obtained from true user behavior as in [41, 37].

Suitable datasets to evaluate recommendation systems should (at least) be composed of logs of recommendations together with their rewards (e.g. a movie and a rating, a music and its listening time, a product and its purchase...). In our case, recommendations are list of items and their positions and rewards are the list of clicks. To the best of our

knowledge, there exist only two public datasets which answer these criteria. [3] First, these two datasets (Yandex then KDD) are described in terms of collected information and parameters inference. Lastly, the choice of parameters for the purely simulated settings are given.

Yandex Dataset

The *Yandex*[4] dataset contains 65 million search queries and 167 million hits, expressing about a month of search activities from users of Yandex search engine. For each query, the user is shown 10 items from a larger set of possibilities defined for each query. Each items are displayed at positions 1 to 10 and the search engine records each click of the user. As in [43], we select the most frequent queries, and keep the 10 most attractive items to display. The experimental setting includes the 10 most frequent queries and requires ranked recommendations of 5 items. To have various settings based on this dataset, we proceed similarly to [43]. The results is averaged on the 10 most frequent queries. These L items are chosen among the most attractive ones selected among all items possible for each query. Then, we observe the regret over the top $K = 5$ positions. We also perform various experimental plan in order to test various values of L. Having various values L allow us to test the impact of the quantity of information given to each algorithm. This leads to a dataset where each entry correspond to a session ID, a query, a list of 10 items displayed and a list of $\{0, 1\}$ corresponding to the user's click. For instance, id000 with the query q has been shown $(item_A, item_B, item_C)$ and gives as feedbacks $(1, 0, 0)$ meaning that during session id000 which is associated with the query q, the user clicked on $item_A$ but not on $item_B$ nor $item_C$.

To compute the PBM parameters, these entries are transposed to q matrices, $\boldsymbol{M}^{[q]}$, (one for each query) by putting each available position in rows, items in columns and each element $\boldsymbol{M}^{[q]}[k, i]$ being the proportion of clicks on item i while being at position k when the query is q. $\boldsymbol{M}^{[q]}$ is constructed by filtering on q and then iterating on each element of the lists of displayed items and their associated list of clicks.

In the Yandex dataset, for each query q, the parameters $(\boldsymbol{\theta}^{[q]}, \boldsymbol{\kappa}^{[q]})$ of PBM are set from the *Singular Value Decomposition* (SVD) of the matrix $\boldsymbol{M}^{[q]} \in \mathbb{R}^{L \times K}$ which contains

3. Event if this work results was made in the context of a company, the data collected so far on Louis Vuitton' websites are not suitable for our experiments due to the lack of recorded logs of past recommendations and due to static nature of these recommendations.

4. Yandex personalized web search challenge, 2013. https://www.kaggle.com/c/yandex-personalized-web-search-challenge

the probability to be clicked for each item in each position. By denoting $\zeta^{[q]}$, the greatest singular value of $\boldsymbol{M}^{[q]}$, and $\boldsymbol{u}^{[q]}$ (respectively $\boldsymbol{v}^{[q]}$) the left (resp. right) singular vector associated to $\zeta^{[q]}$, we set

$$\boldsymbol{\theta}^{[q]} \stackrel{def}{=} \boldsymbol{v}_1^{[q]} \zeta^{[q]} \boldsymbol{u}^{[q]}, \qquad \boldsymbol{\kappa}^{[q]} \stackrel{def}{=} \boldsymbol{v}^{[q]} / \boldsymbol{v}_1^{[q]},$$

such that $\boldsymbol{\kappa}_1^{[q]} = 1$, and ${\boldsymbol{\theta}^{[q]}}^T \boldsymbol{\kappa}^{[q]} = \zeta {\boldsymbol{u}^{[q]}}^T \boldsymbol{v}^{[q]}$. This leads to θ_i values ranging from 0.070 to 0.936, depending on the query, and κ_k values ranging from 0.49 to 1.0,

Note that [41] uses the *expectation-maximization* framework instead of SVD to infer the parameters.

For the Cascading Model, we use the Pyclick [3] package to infer the ($\boldsymbol{\theta}^{[q]}$) parameters. This leads to to θ_i values ranging from 0.0053 to 0.5, depending on the query.

KDD Dataset

The second behavioral dataset is *KDD Cup 2012 track 2*. It consists of session logs of *soso.com*, a Tencent's search engine. It tracks clicks and displays of advertisements on a search engine result web-page, w.r.t. the user query. For each query, at most 3 positions are available for different number of ads to display. Each of the 150M lines contains information about the search (UserId, QueryId...) and the ads displayed (AdId, Position, Click, Impression). We seek the best ads per query, namely the ones with a higher probability to be clicked.

To follow what has been done in previous works, instead of looking for the probability to be clicked per display, we target the probability to be clicked per session. This amounts to discarding the information *Impression*. We also filter the logs to restrict the analysis to (query, ad) couples with enough information: for each query, ads are excluded if they were displayed less than 1,000 times at any of the 3 possible positions. Then, we filter queries that have less than 5 ads satisfying the previous condition. We end up with 8 queries and from 5 to 11 ads per query. In this dataset, each entry is composed of a query and of the couple (item, position) with a boolean indicating if the couple has been clicked on or not. As for Yandex, these entry are transposed to matrices $\boldsymbol{M}^{[q]}$ by iterating on all couples to get proportions of click for each couple.

In this second dataset KDD, for each query q, the parameters are also inferred thanks to SVD for PBM as done for Yandex dataset. This leads to θ_i values ranging from 0.004 to 0.149 and κ_k values ranging from 0.10 to 1.00, depending on the query.

Parameters for other click behavioral models can not be computed due to the structure of this KDD dataset. Indeed, the logs record each couple (item,user) independently, preventing us to have the exhaustive list of items seen by a user at each iteration.

Simulated Data

We choose the value of the PBM parameters $(\boldsymbol{\theta}, \boldsymbol{\kappa})$ in the *purely simulated* setting, to highlight the stability of all the proposed approach even for extreme settings. Namely, we consider $L = 10$ items, $K = 5$ positions, and $\boldsymbol{\kappa} = [1, 0.75, 0.6, 0.3, 0.1]$. The range of values for $\boldsymbol{\theta}$ is either:

— close to zero ($\boldsymbol{\theta}^- = [10^{-3}, 5.10^{-4}, 10^{-4}, 5.10^{-5}, 10^{-5}, 10^{-6}, \ldots, 10^{-6}]$),

— close to one ($\boldsymbol{\theta}^+ = [0.99, 0.95, 0.9, 0.85, 0.8, 0.75, \ldots, 0.75]$),

— characteristic of the one encountered in website interactions such as product discovery on a commercial website ($\boldsymbol{\theta}^W = [0.3, 0.2, 0.15, 0.15, 0.15, 0.10, 0.05, 0.05, 0.01, 0.01]$).

1.4 Conclusion

In this chapter, we gave insights on click behavioral model and bandit algorithms to have a better understanding of how we can model our users' interactions and how we can handle the interactive nature of recommendation systems. To make multi-proposition recommendations, we combine these two concepts in order to dynamically adapt to users' behaviors towards list of recommendations. The next chapter provides state-of-the-art algorithms for this setting.

Chapter 2

Related work

Contents

2.1 Bandits on PBM 43
2.1.1 PMED 44
2.1.2 Focus on (KL)CombUCB1 45
2.2 Bandits on other click behavioral models 47
2.2.1 Focus on TopRank 47
2.3 Related algorithms 49
2.3.1 Focus on OSUB 49
2.4 Conclusion 51

This chapter provides a literature study of the most relevant research works in the topics introduced in the previous chapter. It focuses on algorithms and it is organised according to the goal of each presented algorithm. First, we give an overview of bandit algorithms which aim at recommending lists of items in the Position-based Model (PBM). Then, we extend this literature study to algorithms designed for other click behavioral models, such as Cascading models, or for unspecified settings. Finally, we refer to unimodal bandits, a type of bandits which was not designed for list recommendations but whose mechanism is close to our contribution.

2.1 Bandits on PBM

As presented in Section 1.1.1, PBM [55, 16] relies on two vectors of parameters: $\boldsymbol{\theta} \in [0,1]^L$ and $\boldsymbol{\kappa} \in [0,1]^K$, where θ_i is the probability for the user to click on item i when she/he observes this item, and κ_k is the probability for the user to observe position k. PBM is transposed to the bandit framework in [36, 41, 37]. [36] and [41] propose two approaches based on a Thompson sampling framework, with two different sampling strategies. [41] also introduces several approaches based on the *optimism in face of uncertainty* principle

[5]. However, approaches in [36, 41] assume $\boldsymbol{\kappa}$ known beforehand. In this section, we mainly focus on the PMED algorithm [37] as it does very few additional assumptions on PBM compared to bandits based on Thompson sampling presented here. Then, we also describe the adaptation of CombUCB1 [40] to PBM.

2.1.1 PMED

[37] proposes PMED, an approach to learn both $\boldsymbol{\theta}$ and $\boldsymbol{\kappa}$ while recommending. However, PMED requires the κ_k values to be sorted in decreasing order which is, as discussed below, not a minor assumption.

This algorithm bases its recommendations on the optimal amount of exploration steps needed to have a consistent conclusion on the performance, on each couple item-position (i, k), noted $\{N_{i,k}\}_{i\in[L],k\in[K]}$. To compute exactly this quantity, one needs the true value of $(\boldsymbol{\theta}, \boldsymbol{\kappa})$. As this algorithm faces PBM under uncertainty of position bias, it computes an approximation of this optimal amount of exploration, noted $\{\tilde{N}_{i,l}\}_{i\in[L],k\in[K]}$, by using $(\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\kappa}})$ which are the maximum likelihood estimators of $(\boldsymbol{\theta}, \boldsymbol{\kappa})$. It leads to an optimal regret lower bound.

More precisely, PMED is based on the construction of permutations (referred to as L-allocations in [37]). We remind that a permutation is a list of K elements, from our set of L possible items, in which the position in the list corresponds to its position in the recommendation. For example, with $K = 5$ and $L = 10$, a possible permutation is (7,3,5,1,4) corresponding to putting item 7 in position 1, item 3 in position 2, item 5 in position 3, item 1 in position 4 and item 4 in position 5. PMED then selects an arm among a set of possible permutations noted L_C. PMED loops on this set of permutations in order to gradually collect information on each items and positions and build the next set of permutations noted L_N which will be transferred to L_C as soon as $L_C = \emptyset$. L_C initially contains all (circular) permutations of the list of K elements. PMED loops on L_C until the iteration constrain is reached (meaning $t > T$). It is possible to do so since L_N is never empty. PMED has three ways to add elements to L_N:

— uniform exploration over the pairs (i, k). This exploration stabilizes the estimators.

— exploration based on the estimated optimal amount of exploration steps, $\{\tilde{N}_{i,k}\}_{i\in[L],k\in[K]}$, to collect enough information on items and positions.

— exploitation, which is used when the optimized exploration phase is completed.

An alternative version of PMED, named PMED-Hinge, adds an extra exploration step in order to handle small but not fully converged divergences between arms. This

additional exploration step is helpful to prove the matching between the upper regret bound of this algorithm and the expected regret lower bound proven in [37]. This bound is in $\mathcal{O}(c^*(\boldsymbol{\theta}, \boldsymbol{\kappa}) \log T)$, where $c^*(\boldsymbol{\theta}, \boldsymbol{\kappa})$ only depends on $\boldsymbol{\theta}$ and $\boldsymbol{\kappa}$ and is asymptotically optimal in this setting. Unfortunately, to the best of our knowledge, there is no known closed-form for $c^*(\boldsymbol{\theta}, \boldsymbol{\kappa})$, which hinders the comparison to other algorithms. PMED is not based on Thompson Sampling, even so approaches based on Thompson Sampling are known to deliver more accurate recommendations [7, 2, 35, 11]. PMED is compared to Thompson Sampling approaches which uses approximated law. Thus we can wonder if Thompson Sampling approaches based on the actual reward law could perform better. Moreover, PMED is a complex algorithm to implement and computationally costly given its unstable loops and the matrix decomposition used to build list of items from the permutation matrix.

2.1.2 Focus on (KL)CombUCB1

Combinatorial bandits are also good leads to tackle PBM. In this section, we will describes how the work of [40] can be extended to the PBM setting. Some information about this algorithm have already been given in Section 1.2.3.

Before presenting the way to adapt combinatorial bandits to PBM, we will present a boosted version of CombUCB1 [40]. As mention in Section 1.2.3, among the way to build the upper confidence bound over the expected reward, it is possible to use the Kullback-Leibler (KL) divergence. When it is well applied, using the KL divergence leads to better performance in terms of the quality of the recommendations (the regret decrease faster).

We adapt CombUCB1 to use a KL-based bound and this helps the algorithm to reach a lower regret. This variation on CombUCB1, called KL-CombUCB, chooses its next arm based on a Kullback-Leibler index (see Equation 1.4) instead of the usual confidence upper-bound derived from the Hoeffding's inequality. KL-CombUCB assumes that the weight-vector $\boldsymbol{w}(t)$ is in $\{0,1\}^E$ whereas unspecified combinatorial bandits can have its weight-vector in $\mathbb{R}$ as a weight-vector translate the constraints of the combinatorial settings which have been taken.

We denote $\Delta_{\boldsymbol{a}} \overset{def}{=} \mu^* - \mu_{\boldsymbol{a}}$ the gap between the best expected reward and the reward of an arm $\boldsymbol{a}$, and $\Delta_{min} \overset{def}{=} \min_{\boldsymbol{a} \in \mathcal{A}: \Delta_{\boldsymbol{a}} > 0} \Delta_{\boldsymbol{a}}$ the smallest gap with a sub optimal arm. Kveton et al. proves that the regret of CombUCB1 is upper-bounded by $\mathcal{O}\left(\frac{|E| * K}{\Delta_{min} \log T}\right)$. A similar proof would lead to the same upper-bound for KL-CombUCB.

As mentioned in Section 1.2.4, combinatorial algorithms can also handle PBM. To

Algorithm 3 KL-ComUCB1 (applied to PBM)

Require: number of items L, number of positions K
  for $t = 1, 2, \dots, L$ **do**
    recommend $\boldsymbol{a}(t) = (((t-1)\%L) + 1, (t\%L) + 1, \dots, ((t+K-2)\%L) + 1)$
    observe the clicks-vector $\boldsymbol{c}(t)$
  end for
  for $t = L+1, L+2, \dots$ **do**
    recommend $\boldsymbol{a}(t) = \underset{\boldsymbol{a} \in \mathcal{P}_K^L}{\operatorname{argmax}} \sum_{k=1}^{K} b_{\boldsymbol{a}_k,k}(t)$
    observe the clicks-vector $\boldsymbol{c}(t)$
  end for

apply KL-CombUCB to PBM, we choose the *ground set* $E = [L] \times [K]$, the *feasible set* $\Theta = \{\{(\boldsymbol{a}_k, k) : k \in [K]\} : \boldsymbol{a} \in \mathcal{P}_K^L\}$, and the *expected weights* $\rho_{(i,k)} = \theta_i \kappa_k$ for any "element" $(i,k) \in E$. Note that the observed weights of the generic setting correspond to the clicks-vector in the PBM setting.

The corresponding algorithm, depicted by Algorithm 3, recommends at each iteration t the best permutation given the maximal indices $b_{i,k}(t)$, defined as:

$$b_{i,k}(t) \overset{def}{=} f\left(\hat{\rho}_{i,k}(t), T_{i,k}(t), t\right),$$

where

— $T_{i,k}(t) \overset{def}{=} \sum_{s=1}^{t-1} \mathbb{1}\{\boldsymbol{a}_k(s) = i\}$,
— $\hat{\rho}_{i,k}(t) \overset{def}{=} \frac{1}{T_{i,k}(t)} \sum_{s=1}^{t-1} \mathbb{1}\{\boldsymbol{a}_k(s) = i\} c_k(s)$.

Remark 2. Adapting CombUCB1 to PBM leads to the same algorithm with $b_{i,k}(t)$ being replaced by an upper confidence bound $U_t(i,k) = \hat{\rho}_{i,k}(t-1) + \sqrt{\frac{1.5 \log(t-1)}{T_{i,k}(t-1)}}$.

This optimization problem is a *linear sum assignment problem* which is solvable in $\mathcal{O}\left(K^2(L + \log K)\right)$ time [54]. KL-CombUCB1 [40] applied to PBM leads to an algorithm which suffers a $\mathcal{O}(LK^2/\Delta \log T)$ regret.

Overall, combinatorial bandits and especially KL-CombUCB are fitted to handle PBM settings. Associating items to positions is a combinatorial problem under constraints of unicity of the use of items and positions.

2.2 Bandits on other click behavioral models

Besides PBM, there exists other click behavioral models such as the Cascading model or the Dependent Click model, presented in Section 1.1.2 and 1.1.3. For instance, the cascading model has been extensively studied within the bandit framework [67, 33, 38, 39, 46, 15, 11]. However, the assumption of the cascading model regarding the order of observations is irrelevant when considering items spread in a webpage or items arranged by rows. Due to various user behaviors and industrial needs in terms of display, different bandit algorithms exist to address all these models with less assumptions. To the best of our knowledge, only the algorithms BatchRank [66], TopRank [43], and BubbleRank [44] handle users following a general model covering several behaviors models (and, in particular, PBM). These three algorithms exhibit a regret upper-bound for T consecutive recommendations of at least $\mathcal{O}(\frac{L*K}{\Delta \log T})$, where Δ depends on the so-called *attraction probability* of items. As TopRank presents the best results in terms of regret and complexity, we focus on this method in the following.

2.2.1 Focus on TopRank

To handle various click behavioral models, [43] makes a set of assumptions to define the set of click behavioral models it can handle and ensure that information collected are consistent with its learning process. These assumptions are defined upon $v(\boldsymbol{a}, k)$, which is an unknown function giving the probability that the user clicks on position k at iteration t given the recommendation $\boldsymbol{a} \in \mathcal{A}$, with $\mathcal{A}$ the set of arms. The function v is defined to give a more generic definition of the click behavioral models handled by TopRank. v can be written as $v(\boldsymbol{a}, k) = \chi(\boldsymbol{a}, k)\alpha(\boldsymbol{a}_k)$, with $\chi(\boldsymbol{a}, k)$ the probability that the user examines position k given ranking $\boldsymbol{a}$ and α being an attractiveness function [1]. To follow [43] notations, we note $\boldsymbol{a}^{-1}(i)$ the position of item i in the recommendation $\boldsymbol{a}$. For example, given the recommendation $\boldsymbol{a}$ with $\boldsymbol{a}_k = i$, we have with this notation $\boldsymbol{a}^{-1}(i) = k$. We remind TopRank's assumptions on users' behaviors here:

— $v(\boldsymbol{a}, k) = 0$ for all $k > K$,

— Let $\boldsymbol{a}^* \in \mathcal{A}$ be an optimal action. Then $\max_{a \in \mathcal{A}} \sum_{k=1}^{K} v(\boldsymbol{a}, k) = \sum_{k=1}^{K} v(\boldsymbol{a}^*, k)$,

— Let i and j be items with $\alpha(i) \geq \alpha(j)$ and let $\sigma : \mathcal{A} \to \mathcal{A}$ be the permutation that exchange i and j and leaves other items unchanged. Then for any action $\boldsymbol{a}_t \in \mathcal{A}$, $v(\boldsymbol{a}, \boldsymbol{a}^{-1}(i)) \geq \frac{\alpha(i)}{\alpha(j)} v(\sigma(\boldsymbol{a}), \boldsymbol{a}^{-1}(i))$,

1. Note that we recover the model introduced in Chapter 1, with $v(\boldsymbol{a}, k) = \mathbb{P}(c_k|\boldsymbol{a})$ and $\alpha(\boldsymbol{a}_k) = \theta_{\boldsymbol{a}_k}$.

Algorithm 4 TopRank

$G_1 \leftarrow \emptyset$ and $c \leftarrow \frac{4\sqrt{2/\pi}}{erf(\sqrt{2})}$
for $t = 1, 2, \ldots, T$ **do**
  {Construction of the partition}
  $d \leftarrow 0$
  while $[L] \backslash \bigcup_{c=1}^{d} \mathcal{P}_c^t \neq \emptyset$ **do**
    $d \leftarrow d + 1$
    $\mathcal{P}_d^t \leftarrow \min_{G_t} \left([L] \backslash \bigcup_{c=1}^{d-1} \mathcal{P}_c^t \right)$
  end while
  {Choice of action}
  Recommend $\boldsymbol{a}_t$ uniformly at random from $\mathcal{A}(\mathcal{P}_1^t, \ldots, \mathcal{P}_d^t)$
  Observe the reward $\boldsymbol{c}(t)$
  {Construction of the oriented Graph}
  for $(i, j) \in [L]^2$ **do**
    $U_{tij} \leftarrow \begin{cases} c_i(t) - c_j(t) & \text{, if } i, j \in \mathcal{P}_d^t \text{ for some } d, \\ 0 & \text{, otherwise} \end{cases}$
    $S_{tij} \leftarrow \sum_{s=1}^{t} U_{sij}$ and $N_{tij} \leftarrow \sum_{s=1}^{t} \mid U_{sij} \mid$
    $G_{t+1} \leftarrow G_t \bigcup \left\{ (j, i) : S_{tij} \geq \sqrt{2N_{tij} \log(\frac{c}{\delta}\sqrt{N_{tij}})} \text{ and } N_{tij} > 0 \right\}$
  end for
end for

— For any action $\boldsymbol{a}$ and optimal action $\boldsymbol{a}^*$ with $\alpha(a(k)) = \alpha(\boldsymbol{a}^*(k))$ it holds that $v(\boldsymbol{a}, k) \geq v(\boldsymbol{a}^*, k)$.

Algorithm 4 presents TopRank. In order to propose an optimal ordered list, TopRank is based on three components, each component being updated at each iteration t: a representation of relation between items based on performances, noted G_t, a partition of items based on these relations, noted $\mathcal{P}^t$ and a subset of the possible action raising from this partition, noted $\mathcal{A}(\mathcal{P})$. First TopRank identifies relations between items and represents these relations in the oriented graph G_t, where $G_1 = \emptyset$. More precisely, G_t links two items with the oriented edge (j, i) if TopRank determines with high probability that i has a greater expected reward than j. Formally, G_t is updated at each iteration t with the couple (j, i) if $S_{tij} \geq \sqrt{2N_{tij} \log(\frac{c}{\delta}\sqrt{N_{tij}})}$ and $N_{tij} > 0$, where S_{tij} is the number of clicks on i rather than j and N_{tij} is the number of times i and j where simultaneously presented. Then, at each iteration t, TopRank computes a partition of $[L]$ into $\mathcal{P}_1^t, \ldots, \mathcal{P}_d^t$. This partition is made by extracting items which are minimum according to G_t. For $X \subseteq [L]$, the minimum according to G_t is $\min_{G_t}(X) \stackrel{def}{=} \{i \in X : (i, j) \notin G_t \text{ for all } j \in X\}$. This partition

splits all elements of $[L]$ into exclusive groups where all elements have the same relative information, ie any $j \in \mathcal{P}_{c+1}^t$ leads with high probability to a lower expected reward than any $i \in \mathcal{P}_c^t$. This partition is key to construct the set of actions $\mathcal{A}(\mathcal{P}_1^t, \ldots, \mathcal{P}_d^t)$, where an action $\boldsymbol{a}$ places items in $\mathcal{P}_1^t$ in the first $\mid \mathcal{P}_1^t \mid$ positions, the items in $\mathcal{P}_2^t$ in the next $\mid \mathcal{P}_2^t \mid$ positions, and so one. In each subset of a partition, the items are randomized. Formally,

$$\mathcal{A}(\mathcal{P}_1^t, \ldots, \mathcal{P}_d^t) \stackrel{def}{=} \{a \in \mathcal{A} : \max_{i \in \mathcal{P}_c^t} \boldsymbol{a}^{-1}(i) \leq \min_{i \in \mathcal{P}_{c+1}^t} \boldsymbol{a}^{-1}(i) \text{ for all } c \in [d-1]\}.$$

Thus $\mathcal{A}(\mathcal{P}_1^t, \ldots, \mathcal{P}_d^t)$ is a subset of $\mathcal{A}$ containing all the arms which are compliant with the partial order identified by the partition $\mathcal{P}_1^t, \ldots, \mathcal{P}_d^t$ and from which TopRank chooses its next recommendation.

As said in the introduction of this section, TopRank exhibits a regret upper-bound for T iterations of $\mathcal{O}(LK/\Delta \log T)$, where Δ depends on the attraction probability of items. TopRank is the first efficient algorithm to handle several click behavioral models, without strong assumptions.

2.3 Related algorithms

Bandits aim to choose actions one at a time. These actions can take various forms. Since in our setting, our actions are totally ordered lists of items, the two first sections of this chapter described works related to similar settings, where click behavioral model assumptions are made to help bandits being more efficient. Nevertheless, among the contributions presented in this thesis, some of them can be brought closer to works handling slightly different settings. This section will present such approaches. More precisely, we will focus on OSUB, a unimodal bandit for single recommendation tasks. The work presented in this thesis is partly an extension of this algorithm.

2.3.1 Focus on OSUB

The contributions presented in Chapter 3, and 5 extend the *unimodal* bandit setting [13] which assumes the existence of a known graph $G = (V, E)$ carrying a partial order on the set of bandit arms denoted $\mathcal{A}$. Unimodal bandit algorithms are aware of G, but ignore the partial order induced by the edges of G. However, they rely on G to efficiently browse the arms up to the best one.

Typically, OSUB algorithm [13] selects at each iteration t, an arm $\boldsymbol{a}(t)$ in the neighborhood $\mathcal{N}_G(\tilde{\boldsymbol{a}}(t))$ given G of the current best arm $\tilde{\boldsymbol{a}}(t)$ (a.k.a. the *leader*).

Unimodal bandit algorithms are based on a specific assumption which links the set of possible actions and gives the general structure of G. Let us first recall the definition of unimodality as described in [13] and then detail OSUB algorithm.

Definition 1 (Unimodality)**.** *Let $\mathcal{A}$ be a set of arms, and $(\nu_{\boldsymbol{a}})_{\boldsymbol{a}\in\mathcal{A}}$ a set of reward distributions of respective expectations $(\mu_{\boldsymbol{a}})_{\boldsymbol{a}\in\mathcal{A}}$. Let $G = (V, E)$ be an undirected graph with vertices $V = \mathcal{A}$ and edges $E \subseteq V^2$. The set of expected rewards $(\mu_{\boldsymbol{a}})_{\boldsymbol{a}\in\mathcal{A}}$ is* unimodal w.r.t. *G, if and only if:*

1. *the set of expected rewards admits a unique best arm:* $\operatorname{argmax}_{\boldsymbol{a}\in\mathcal{A}} \mu_{\boldsymbol{a}} = \{\boldsymbol{a}^*\}$*;*
2. *and from any arm $\boldsymbol{a} \neq \boldsymbol{a}^*$, there exists a path $(\boldsymbol{a}^0, \boldsymbol{a}^1, \ldots, \boldsymbol{a}^n)$ in G such that $\boldsymbol{a}^0 = \boldsymbol{a}$, $\boldsymbol{a}^n = \boldsymbol{a}^*$, and $\forall i \in [n], \mu_{\boldsymbol{a}^i} > \mu_{\boldsymbol{a}^{i-1}}$.*

Note that the second property of unimodal sets of expected rewards is equivalent to the property stating that *from any sub-optimal arm $\boldsymbol{a}$, there exists an arm $\boldsymbol{a}' \in \mathcal{N}_G(\boldsymbol{a})$ such that $\mu_{\boldsymbol{a}'} > \mu_{\boldsymbol{a}}$*, where $\mathcal{N}_G(\boldsymbol{a})$ is the neighborhood of $\boldsymbol{a}$ in G.

These assumptions structure G and allow OSUB to work as follows (see pseudo code in Algorithm 5). OSUB uses an index based on KL-divergence upper confidence bounds defined for each arm. This index's definition comes from KL-UCB algorithm [22] and aims to an optimistic exploration. A similar definition as the one provided in Section 1.2.3 leads to define the index as:

$$b_k(t) \overset{def}{=} f\left(\hat{\mu}_k(t), T_k(t), \tilde{T}_{\tilde{\boldsymbol{a}}(t)}(t) + 1\right),$$

where

— $T_k(t) \overset{def}{=} \sum_{s=1}^{t-1} \mathbb{1}\{\boldsymbol{a}_k(s) = i\}$,

— $\hat{\mu}_k(t) \overset{def}{=} \frac{1}{T_k(t)} \sum_{s=1}^{t-1} \mathbb{1}\{\boldsymbol{a}_k(s) = i\} c_k(s)$,

— $\tilde{\boldsymbol{a}}(t)$ the *leader*, i.e. the recommendation with the best *pseudo average reward* $\hat{\mu}_k(t)$,

— $\tilde{T}_{\tilde{\boldsymbol{a}}(t)}(t) \overset{def}{=} \sum_{s=1}^{t-1} \mathbb{1}\{\tilde{\boldsymbol{a}}(s) = \boldsymbol{a}\}$.

OSUB searches in the neighborhood of the current leader $\tilde{\boldsymbol{a}}(t)$. It chooses the next action $\boldsymbol{a}$ such as $\boldsymbol{a} = \operatorname{argmax}_{\boldsymbol{a}\in\mathcal{N}_G(\tilde{\boldsymbol{a}})} b_{\boldsymbol{a}}(t)$ with $\mathcal{N}_G(\boldsymbol{a}) = \{l : (l, \boldsymbol{a}) \in E\} \cup \{\boldsymbol{a}\}.n$

In this algorithm, ties are broken arbitrarily and the leader $\tilde{\boldsymbol{a}}(t)$ is often recommended in order to ensure that the number of times an arm has been selected is at least proportional to the number of times it has been the leader. This can be removed but it

Algorithm 5 OSUB

Require: graph $G = (V, E)$
Require: maximal degree of nodes in G γ
for $t \geq 1$ **do**
select

$$k(n) = \begin{cases} \tilde{\boldsymbol{a}}(t) & \text{, if } \frac{\tilde{T}_{\tilde{\boldsymbol{a}}(t)}(t)}{\gamma+1} \in \mathbb{N}, \\ \underset{\substack{k \in \{\tilde{\boldsymbol{a}}(t)\} \\ \cup \mathcal{N}_{\tilde{\boldsymbol{\pi}}}(\tilde{\boldsymbol{a}}(t))}}{\operatorname{argmax}} \sum_{k=1}^{K} b_k(t) & \text{, otherwise} \end{cases}$$

observe the clicks-vector $\boldsymbol{c}(t)$
end for

simplifies the regret analysis. By restricting the exploration to this neighborhood, the regret suffered by OSUB scales as $\mathcal{O}(\gamma/\Delta \log T)$, where γ is the maximum degree of G and $\Delta = \min_{\boldsymbol{a} \in \mathcal{N}_G(\boldsymbol{a}^*)} \mu^* - \mu_{\boldsymbol{a}}$. This bound has to be compared with $\mathcal{O}(|\mathcal{A}|/\Delta \log T)$ if the arms are independent.

Remark 3 (Rank-1-Bandit)**.** [34, 32] propose *Rank-1-Bandit*, an algorithm which apply OSUB on a similar framework as ours. In their framework, the probability for an item i to be clicked in position k is also $\boldsymbol{\theta}_i \boldsymbol{\kappa}_k$. The difference lies in the recommendation setting. While in our case the algorithm has to choose an item for each K positions at each iteration, in the rank-1 bandit setting, the algorithm only picks one item and pairs it with the most appropriate position to be displayed.

2.4 Conclusion

This overview of state-of-the-art algorithms addressing the bandit setting under a PBM assumption shows that very few of these algorithms address this model with both $(\boldsymbol{\theta}, \boldsymbol{\kappa})$ unknown (fully unknown PBM). The only approach is PMED which is computationally costly. We aim at developing efficient algorithms both in terms of computational time and regret performance. To do so, on the one hand we extend unimodal bandits (presented in Section 2.3) to reach an efficient algorithm and prove its regret bound. On the other hand, in Section 1.3 we approximate draws given the a posteriori law on the parameters of the fully unknown PBM and we use these draws in the Thompson Sampling framework.

Furthermore, many click behavioral models exist to address various recommendation settings. To be more generic, some algorithms address simultaneously several click be-

havioral models, when they fall in a set of reasonable assumptions. We contribute to this area with a unimodal bandit which is able to address click behavioral models such as CM and PBM simultaneously. This contribution leads to decrease the number of assumptions needed for generic list recommendation, down to the existence of an identifiable total order on the items. We also decrease the expected regret lower bound identified by TopRank for this setting. Even if unimodal approaches were not initially designed to tackle our settings, we extend these approaches to our tasks.

Chapter 3

Unimodal bandit for the position-based model

Contents

3.1 Relation with unimodality 54
3.2 Parametric graph for unimodal ranking bandit 57
3.3 Theoretical analysis 59
3.3.1 Discussion 63
3.4 Practical results 65
3.5 Conclusion 67

The idea presented in this chapter is based on the mapping of lists of recommendable items into a unimodal graph. We elaborate on this idea: how PBM can be adapted to a unimodal graph? How can a bandit algorithm explore this graph to find the best possible list? The contributions presented in this chapter have been published in [26] at ICML'21.

We tackle a position-based online learning to rank (PB-OLR) bandit setting, which is defined in Section 1.3 and covers PBM click model, with an unimodal bandits *point of view [13]. First, we expose a family of parametric graphs of degree $L-1$ over permutations, such that the PBM setting is unimodal w.r.t. one graph in this family. While the corresponding graph is unknown from the learner, graphs of this family enable an efficient exploration strategy of the set of potential recommendations. Secondly, we introduce a new bandit algorithm, GRAB, which learns online the appropriate graph in this family and bases its recommendations on the learned graph. From an application point of view, this algorithm has several interesting features: it is simple to implement and efficient in terms of computation time; it handles the PBM bandit setting without any knowledge on the impact of positions (contrarily to many competitors presented in Chapter 2.1); and it empirically exhibits a regret on par with other theoretically proven algorithms on both artificial and real datasets. In particular, we prove a $O(L/\Delta \log T)$ regret upper-bound for*

GRAB. Regret upper-bounds give guaranties on the performance of algorithms in the worst case scenario. In order to give more context to this theoretical result, a comparison of the assumptions and the regret upper-bounds of related algorithms is shown in Table 3.1. This Table shows that GRAB achieves one of the lowest regret bound with fewer information. The lowest regret bound is achieved by PBM-PIE [41] which needs the value of $\boldsymbol{\kappa}$. As discuss in Section 2.1.1, PMED-Hinge [37] has a bound defined on $c^*(\boldsymbol{\theta}, \boldsymbol{\kappa})$ which makes the comparison to other regret bound difficult. The proof of our bound extends OSUB's proof [13] both (i) to the context of a graph learned online, and (ii) to the combinatorial semi-bandit setting.

This chapter is organised as follow: Section 3.1 describes the assumption taken to use unimodality bandits in our setting. Section 3.2 presents GRAB. Finally theoretical and practical results are presented in Section 3.3 and 3.4. We conclude in Section 3.5

Table 3.1: Settings and upper-bound on cumulative regret for state-of-the-art algorithms. $\mathcal{N}_{\pi^*}(\boldsymbol{a}^*)$ is a set of recommendations in the neighborhood of the best recommendation. K_{max} is the maximum number of differences between two arms; see. Theorem 2 for a specific definition.

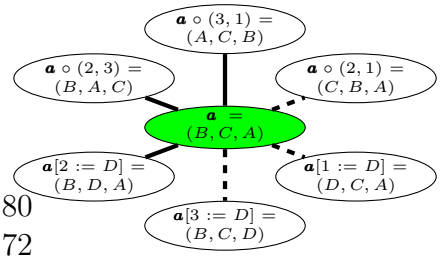
Algorithm	**Handled settings**	**Regret**	**Δ, assuming $\theta_1 \geqslant \theta_2 \geqslant \cdots \geqslant \theta_L$**
GRAB (our algorithm)	PBM	$\mathcal{O}\left(\frac{L}{\Delta}\log T\right)$	$\min_{\boldsymbol{a}\in\mathcal{N}_{\pi^*}(\boldsymbol{a}^*)} \mu^* - \mu_{\boldsymbol{a}}$
CombUCB1 [40]	PBM	$\mathcal{O}\left(\frac{LK^2}{\Delta}\log T\right)$	$\min_{\boldsymbol{a}\in\mathcal{P}_K^L} \mu^* - \mu_{\boldsymbol{a}}$
PBM-PIE [41]	PBM with $\boldsymbol{\kappa}$ known	$\mathcal{O}\left(\frac{L-K}{\Delta}\log T\right)$	$\min_{i\in\{K+1,\dots,L\}} \mu^* - \mu_{\boldsymbol{a}[K:=i]}$
PMED-Hinge [37]	PBM with $\kappa_1 \geqslant \cdots \geqslant \kappa_K$	$\mathcal{O}\left(c^*(\boldsymbol{\theta}, \boldsymbol{\kappa})\log T\right)$	$\varnothing$
TopRank [43]	PBM with $\kappa_1 \geqslant \cdots \geqslant \kappa_K$, CM, ...	$\mathcal{O}\left(\frac{LK}{\Delta}\log T\right)$	$\min_{(j,i)\in[L]\times[K]:j>i} \frac{\theta_i - \theta_j}{\theta_i}$
OSUB [13]	Unimodal	$\mathcal{O}\left(\frac{\gamma}{\Delta}\log T\right)$	$\min_{\boldsymbol{a}\in\mathcal{N}_G(\boldsymbol{a}^*)} \mu^* - \mu_{\boldsymbol{a}}$
KL-CombUCB (Theorem 2)	Combinatorial	$\mathcal{O}\left(\frac{\lvert\mathcal{A}\rvert K_{max}^2}{\Delta}\log T\right)$	$\min_{\boldsymbol{a}\in\mathcal{A}} \mu^* - \mu_{\boldsymbol{a}}$

3.1 Relation with unimodality

The setting tackled in this Chapter is the PB-OLR setting (see Section 1.3). It assumes that clicks are independent. Apart from this global assumption, the proposed algorithm

$$\boldsymbol{\rho} = \begin{array}{c} \\ A \\ B \\ C \\ D \end{array} \begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{bmatrix} 1.00 & 0.90 & 0.80 \\ 0.90 & 0.81 & 0.72 \\ 0.80 & 0.72 & 0.64 \\ 0.70 & 0.63 & 0.56 \end{bmatrix} \end{array}$$

$\boldsymbol{a} = (B, C, A)$
$\boldsymbol{\pi} = (1, 3, 2)$
$\Pi_{\boldsymbol{\rho}}(\boldsymbol{a}) = \{\boldsymbol{\pi}\}$



$$\rho_{a_1,1} = 0.90 \geqslant \rho_{a_2,2} = 0.72 < \rho_{a_3,3} = 0.80$$
$$\rho_{a_{\pi_1},\pi_1} = 0.90 \geqslant \rho_{a_{\pi_2},\pi_2} = 0.80 \geqslant \rho_{a_{\pi_3},\pi_3} = 0.72$$
$$\mu_{\boldsymbol{a}\circ(3,1)} = 1.00{+}0.72{+}0.72 = 2.44 > \mu_{\boldsymbol{a}} = 0.90{+}0.72{+}0.80 = 2.42$$

Figure 3.1: Assumption 1 in practice. To distinguish between items and positions, the 4 items are denoted A, B, C, and D. **On the left**: parameters and considered recommendation $\boldsymbol{a}$. We consider a matrix of probabilities of clicks $\boldsymbol{\rho}$ which corresponds to a PBM click model, and a sub-optimal recommendation $\boldsymbol{a}$. The corresponding set $\Pi_{\boldsymbol{\rho}}(\boldsymbol{a})$ of appropriate rankings of positions is composed of a unique permutation $\boldsymbol{\pi}$. **On the right**: corresponding neighborhoods. Solid lines identify the neighborhood $\mathcal{N}_{\boldsymbol{\pi}}(\boldsymbol{a})$ used by GRAB, and both solid and dashed lines correspond to the neighborhood $\mathcal{N}_G(\boldsymbol{a})$ used by S-GRAB (see the appendix A.7 for details regarding S-GRAB). Note that there is a recommendation better than $\boldsymbol{a}$ in both neighborhoods: $\boldsymbol{a} \circ (3, 1) = (A, C, B)$.

assumes a relaxed version of unimodality. Here we present this assumption and state its relation with PBM. We first define the set of *appropriate rankings of positions*: for each recommendation $\boldsymbol{a} \in \mathcal{P}_K^L$, we denote $\Pi_{\boldsymbol{\rho}}(\boldsymbol{a}) \subseteq \mathcal{P}_K^K$ the set of permutations $\boldsymbol{\pi}$ of the K positions such that $\rho_{a_{\pi_1},\pi_1} \geqslant \rho_{a_{\pi_2},\pi_2} \geqslant \cdots \geqslant \rho_{a_{\pi_K},\pi_K}$, with $\rho_{a_{\pi_1},\pi_1}$ being the probability for a user to click on item placed at position π_1 in the list a as defined in Section 1.3. Therefore, an appropriate ranking of positions orders the positions from the one with the highest probability of click to the one with the lowest probability of click. See Figure 3.1 for an example of $\Pi_{\boldsymbol{\rho}}(\boldsymbol{a})$.

With this notation, our assumption is the following:

Assumption 1 (Relaxed Unimodality)**.** *For any recommendation $\boldsymbol{a} \in \mathcal{P}_K^L$ and any ranking of positions $\boldsymbol{\pi} \in \Pi_{\boldsymbol{\rho}}(\boldsymbol{a})$, if $\mu_{\boldsymbol{a}} \neq \mu^*$, with $\mu_{\boldsymbol{a}}$ defined by Equation* (1.5)*, then either there exists $k \in [K-1]$ such that*

$$\mu_{\boldsymbol{a}} < \mu_{\boldsymbol{a}\circ(\pi_k,\pi_{k+1})} \tag{3.1}$$

or there exists $i \in [L] \setminus \boldsymbol{a}([K])$ such that

$$\mu_{\boldsymbol{a}} < \mu_{\boldsymbol{a}[\pi_K:=i]}, \tag{3.2}$$

where

- $\boldsymbol{a} \circ (\pi_k, \pi_{k+1})$ *is the permutation for which the items at positions* π_k *and* π_{k+1} *are swapped,*
- $\boldsymbol{a}[\pi_K := i]$ *is the permutation which is the same as* $\boldsymbol{a}$ *for any position* $k \neq \pi_K$*, and such that* $\boldsymbol{a}[\pi_K := i]_{\pi_K} = i$*,*
- *and* $\boldsymbol{a}([K])$ *is the set of items recommended by* $\boldsymbol{a}$*, namely* $\boldsymbol{a}([K]) \overset{def}{=} \{a_1, \ldots, a_K\}$.

Assumption 1 relates to a natural property of standard click models: (i) for the optimal recommendation, the position with the k-th highest probability to be observed is the one displaying the k-th most attractive item, (ii) for a sub-optimal recommendation, swapping two consecutive items, given this order, leads to an increase of the expected reward. However, Assumption 1 considers the order based on the click probabilities $\rho_{a_k,k}$, not on the observation probabilities κ_k. Figure 3.1 gives an example of both orders and of the neighborhood associated to the ranking $\boldsymbol{\pi}$ defined after the order on click probabilities $\rho_{a_k,k}$.

Assumption 1 relates to the unimodality of the set of expected rewards $(\mu_{\boldsymbol{a}})_{\boldsymbol{a} \in \mathcal{P}_K^L}$. The definition of unimodality in [13] is recall in Section 2.3.1, Definition 1. The relation between unimodality and Assumption 1 is expresses here .

Let us assume that there exists a unique recommendation $\boldsymbol{a}^*$ with maximum expected reward, and denote $\mathcal{F} = (\boldsymbol{\pi}_{\boldsymbol{a}})_{\boldsymbol{a} \in \mathcal{P}_K^L}$ a set of rankings of positions such that for any recommendation $\boldsymbol{a}$, $\boldsymbol{\pi}_{\boldsymbol{a}} \in \Pi_{\boldsymbol{\rho}}(\boldsymbol{a})$. Then, by denoting $G_{\mathcal{F}} = (V, E_{\mathcal{F}})$ the directed graph with vertices $V = \mathcal{P}_K^L$ and edges

$$E_{\mathcal{F}} \overset{def}{=} \left\{(\boldsymbol{a}, \boldsymbol{a} \circ (\pi_{\boldsymbol{a}k}, \pi_{\boldsymbol{a}(k+1)})) : k \in [K-1]\right\} \cup \left\{(\boldsymbol{a}, \boldsymbol{a}[\pi_{\boldsymbol{a}K} := i]) : i \in [L] \setminus \boldsymbol{a}([K])\right\},$$

$(\mu_{\boldsymbol{a}})_{\boldsymbol{a} \in \mathcal{P}_K^L}$ is unimodal [1] with respect to $G_{\mathcal{F}}$. Note that this graph is unknown from the algorithm as it builds upon the unknown mapping $\mathcal{F}$. However, this mapping may be learned online, paving the way to an OSUB-like algorithm to explore the space of recommendations.

While the existence of a better recommendation in the neighborhood defined given this order is less intuitive, it remains true for state-of-the-art click models (PBM, the cascading model, and the dependent click model) and paves the way to an algorithm based on observed random variables. Note also that while there exists a better recommendation

1. While the definition of unimodality in [13] involves an **undirected** graph, OSUB only requires a **directed** graph and the existence of a strictly increasing path from any sub-optimal arm to the optimal one.

both in the neighborhood based on the order on observation probability and in the neighborhood based on the order on click probability, this is not true for any neighborhood based on any arbitrary order (as soon as $K \geqslant 4$).

Hereafter, Lemma 1, states that Assumption 1 is weaker than the PBM one. The proof of this Lemma is deferred to the appendix.

Lemma 1. *Let $(L, K, (\theta_i \kappa_k)_{(i,k)\in[L]\times[K]})$ be an online learning to rank problem with users following PBM, with positive probabilities of looking at a given position. Then Assumption 1 is fulfilled.*

3.2 Parametric graph for unimodal ranking bandit

Our algorithm, GRAB (for parametric Graph for unimodal RAnking Bandit), takes inspiration from the unimodal bandit algorithm OSUB [13] by selecting at each iteration t an arm $\boldsymbol{a}(t)$ in the neighborhood of the current best arm (a.k.a. the leader), noted $\tilde{\boldsymbol{a}}(t)$. While in OSUB the neighborhood is known beforehand, here we learn it online. GRAB is described in Algorithm 6. This algorithm uses the following notations:

At each iteration t, we denote

$$\hat{\rho}_{i,k}(t) \overset{def}{=} \frac{1}{T_{i,k}(t)} \sum_{s=1}^{t-1} \mathbb{1}\{a_k(s) = i\} c_{a_k(t)}(s),$$

the average number of clicks obtained at position k when displaying item i at this position, where

$$T_{i,k}(t) \overset{def}{=} \sum_{s=1}^{t-1} \mathbb{1}\{a_k(s) = i\}$$

is the number of time item i has been displayed at position k; $\hat{\rho}_{i,k}(t) \overset{def}{=} 0$ when $T_{i,k}(t) = 0$.

The *leader* $\tilde{\boldsymbol{a}}(t)$ is the recommendation with the best *pseudo average reward* $\bar{\mu}_{\boldsymbol{a}}(t) \overset{def}{=} \sum_{k=1}^{K} \hat{\rho}_{a_k,k}(t)$, and we note

$$\tilde{T}_{\boldsymbol{a}}(t) \overset{def}{=} \sum_{s=1}^{t-1} \mathbb{1}\{\tilde{\boldsymbol{a}}(s) = \boldsymbol{a}\},$$

the number of times the leader is $\boldsymbol{a}$ for iterations 1 to $t-1$.

Finally, the statistics $\hat{\rho}_{i,k}(t)$ are paired with their respective *indices*

$$b_{i,k}(t) \overset{def}{=} f\left(\hat{\rho}_{i,k}(t), T_{i,k}(t), \tilde{T}_{\tilde{\boldsymbol{a}}(t)}(t) + 1\right),$$

Algorithm 6 GRAB: parametric Graph for unimodal RAnking Bandit

Require: number of items L, number of positions K

1: **for** $t = 1, 2, \dots$ **do**

2: $\quad \tilde{\boldsymbol{a}}(t) \leftarrow \underset{\boldsymbol{a} \in \mathcal{P}_K^L}{\operatorname{argmax}} \sum_{k=1}^{K} \hat{\rho}_{a_k,k}(t)$

3: $\quad$ find $\tilde{\boldsymbol{\pi}}(t)$ s.t. $\hat{\rho}_{\tilde{a}_{\tilde{\pi}_1(t)}(t),\tilde{\pi}_1(t)}(t) \geqslant \hat{\rho}_{\tilde{a}_{\tilde{\pi}_2(t)}(t),\tilde{\pi}_2(t)}(t) \geqslant \cdots \geqslant \hat{\rho}_{\tilde{a}_{\tilde{\pi}_K(t)}(t),\tilde{\pi}_K(t)}(t)$

4: $\quad$ recommend

$$\boldsymbol{a}(t) = \begin{cases} \tilde{\boldsymbol{a}}(t) & \text{, if } \frac{\tilde{T}_{\tilde{\boldsymbol{a}}(t)}(t)}{L} \in \mathbb{N}, \\ \underset{\substack{\boldsymbol{a} \in \{\tilde{\boldsymbol{a}}(t)\} \\ \cup \mathcal{N}_{\tilde{\boldsymbol{\pi}}}(\tilde{\boldsymbol{a}}(t))}}{\operatorname{argmax}} \sum_{k=1}^{K} b_{a_k,k}(t) & \text{, otherwise} \end{cases}$$

$\quad$ where $\mathcal{N}_{\boldsymbol{\pi}}(\boldsymbol{a}) = \{\boldsymbol{a} \circ (\pi_k, \pi_{k+1}) : k \in [K-1]\} \cup \{\boldsymbol{a}[\pi_K := i] : i \in [L] \setminus \boldsymbol{a}([K])\}$

5: $\quad$ observe the clicks vector $\boldsymbol{c}(t)$

6: **end for**

where $f(\hat{\rho}, s, t)$ stands for

$$\sup\{p \in [\hat{\rho}, 1] : s \times \mathrm{kl}(\hat{\rho}, p) \leq \log(t) + 3\log(\log(t))\},$$

with

$$\mathrm{kl}(p, q) \overset{def}{=} p \log\left(\frac{p}{q}\right) + (1-p)\log\left(\frac{1-p}{1-q}\right),$$

the *Kullback-Leibler divergence* from a Bernoulli distribution of mean p to a Bernoulli distribution of mean q; $f(\hat{\rho}, s, t) \overset{def}{=} 1$ when $\hat{\rho} = 1$, $s = 0$, or $t = 0$.

At each iteration t, GRAB first identifies the leader $\tilde{\boldsymbol{a}}(t)$, and then recommends either $\tilde{\boldsymbol{a}}(t)$ every L-th iteration, or the best permutation in the inferred neighborhood, given the sum of indices $\sum_{k=1}^{K} b_{a_k,k}(t)$ (see Figure 3.1 for an example of a neighborhood). Each time an argmax is computed, the ties are randomly broken.

To finish the presentation of GRAB, let us now discuss its initialisation and its time-complexity.

Remark 4 (Initialisation)**.** The initialisation of the algorithm is handled through the default value of indices $b_{i,k}$: 1. This value ensures that any permutation is recommended at least once, as soon as it belongs to the neighborhood of an arm which is often the leader. If a permutation is not in such neighborhood, the theoretical analysis in Section 3.3 proves that this permutation is sub-optimal, hence it does not matter whether this permutation is explored at least once or not.

Remark 5 (Algorithmic Complexity)**.** Even though the two optimization steps might seem costly, at each iteration the choice of a recommendation is done in a polynomial time w.r.t. L and K: first, the maximization at Line 2 is a *linear sum assignment problem* which is solvable in $\mathcal{O}\left(K^2(L+\log K)\right)$ time [54]; it is not required to scan the $L!/(L-K)!$ permutations of K distinct items among L. Secondly, the maximization at Line 4 is over a set of $L-1$ recommendations and is equivalent to the maximization of

$$B_{\boldsymbol{a}}(t) = \sum_{k=1}^{K} b_{a_k,k}(t) - \sum_{k=1}^{K} b_{\tilde{a}_k(t),k}(t),$$

which reduces to the sum of up to four $b_{a_k,k}(t)$ terms as we are looking at recommendations $\boldsymbol{a}$ in the neighborhood of the leader. Specifically, either

— $\boldsymbol{a} = \tilde{\boldsymbol{a}}(t)$ and $B_{\boldsymbol{a}}(t) = 0$,

— or $\boldsymbol{a} = \tilde{\boldsymbol{a}}(t) \circ (k, k')$ and $B_{\boldsymbol{a}}(t) = b_{\tilde{a}_{k'},k}(t) + b_{\tilde{a}_k,k'}(t) - b_{\tilde{a}_k,k}(t) - b_{\tilde{a}_{k'},k'}(t)$,

— or $\boldsymbol{a} = \tilde{\boldsymbol{a}}(t)[k := i]$ and $B_{\boldsymbol{a}}(t) = b_{i,k}(t) - b_{\tilde{a}_k,k}(t)$.

Hence, this maximization requires $\mathcal{O}(L)$ computation time. Overall, the computation time per iteration is a $\mathcal{O}\left(K^2(L+\log K)\right)$.

3.3 Theoretical analysis

As already mentionned in 2.3.1, the proof of the upper-bound on the regret of GRAB follows a similar path as the proof of OSUB [13]: (1) apply standard bandit analysis to control the regret under the condition that the leader $\tilde{\boldsymbol{a}}(t)$ is the best arm $\boldsymbol{a}^*$, and (2) upper-bound the expected number of iterations such that $\tilde{\boldsymbol{a}}(t) \neq \boldsymbol{a}^*$ by a $\mathcal{O}(\log\log T)$. The inference of the rankings on positions adds up a third step (3) upper-bounding the expected number of iterations such that $\tilde{\boldsymbol{\pi}}(t) \notin \Pi_{\boldsymbol{\rho}}\left(\tilde{\boldsymbol{a}}(t)\right)$ for GRAB.

The first step differs from [13], as we have to account for the semi-bandit feedback. We note that when the leader is the best arm, GRAB behaves as a Kullback-Leibler variation of CombUCB1 [40] that we call KL-CombUCB in the following (see Section 2.1.2 for a complete definition of KL-CombUCB). We derive an upper-bound specific to KL-CombUCB which accounts for the fact that the maximization at Line 4 of Algorithm 6 can be reduced to the maximization over sums of at most 4 terms (see Remark 5). In the context of GRAB, this new result, expressed by Theorem 2, reduces the regret-bound by a factor K w.r.t. the standard upper-bound for CombUCB1.

The second part of the analysis is based on the fact that with high probability

$\bar{\mu}_{\boldsymbol{a}}(t) > \bar{\mu}_{\boldsymbol{a}'}(t)$ if $\mu_{\boldsymbol{a}} > \mu_{\boldsymbol{a}'}$, which derives from the control of the deviation of each $\hat{\rho}_{i,k}(t)$. Here lies the second main difference with Combes and Proutière's analysis: we control the deviation of each individual $\hat{\rho}_{i,k}(t)$ while they control the deviation of $\hat{\mu}_{\boldsymbol{a}}(t) \overset{def}{=} (\sum_{s=1}^{t-1} \mathbb{1}\{\boldsymbol{a}(s) = \boldsymbol{a}\})^{-1} \sum_{s=1}^{t-1} \mathbb{1}\{\boldsymbol{a}(s) = \boldsymbol{a}\} r(s)$. Again, the analysis benefits from the small number of differences between recommendations in the neighborhood of the leader. Moreover, the analysis handles the fact that the neighborhoods may change from an iteration to another, while the neighborhoods are constant in Combes and Proutière's analysis. The corresponding result is expressed, in the following, by Lemma 2.

Finally, the number of iterations at which the inferred ranking on the positions is inappropriate is controlled by Lemma 3. The proof of this lemma is eased by the fact that the number of times the leader is played is at least proportional to the number of times it is the leader.

We now propose and prove the main theorem that upper-bounds the regret of GRAB. Its proof is given after the presentation of all the necessary theorems and lemmas.

Theorem 1 (Upper-Bound on the Regret of GRAB)**.** *Let* $\left(L, K, (\rho_{i,k})_{(i,k)\in[L]\times[K]}\right)$ *be an online learning to rank problem satisfying Assumption 1 and such that there exists a unique recommendation* $\boldsymbol{a}^*$ *with maximum expected reward. When facing this problem, GRAB fulfills:*

$$\forall \boldsymbol{a} \in \mathcal{N}_{\boldsymbol{\pi}^*}(\boldsymbol{a}^*), \quad \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\left\{\tilde{\boldsymbol{a}}(t)=\boldsymbol{a}^*,\ \tilde{\boldsymbol{\pi}}(t)=\boldsymbol{\pi}^*,\ \boldsymbol{a}(t)=\boldsymbol{a}\right\}\right] \leqslant \frac{8}{\Delta_{\boldsymbol{a}}^2} \log T + \mathcal{O}\left(\log\log T\right), \quad (3.3)$$

$$\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{\tilde{\boldsymbol{a}}(t) \neq \boldsymbol{a}^*\}\right] = \mathcal{O}\left(\log\log T\right), \quad (3.4)$$

$$\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{\tilde{\boldsymbol{\pi}}(t) \notin \Pi_{\boldsymbol{\rho}}\left(\tilde{\boldsymbol{a}}(t)\right)\}\right] = \mathcal{O}\left(1\right), \quad (3.5)$$

and hence

$$\begin{aligned} R(T) &\leqslant \sum_{\boldsymbol{a}\in\mathcal{N}_{\boldsymbol{\pi}^*}(\boldsymbol{a}^*)} \frac{8}{\Delta_{\boldsymbol{a}}} \log T + \mathcal{O}\left(\log\log T\right) \\ &= \mathcal{O}\left(\frac{L}{\Delta_{min}} \log(T)\right), \end{aligned} \quad (3.6)$$

where $\boldsymbol{\pi}^*$ *is the unique ranking of positions in* $\Pi_\rho(\boldsymbol{a}^*)$*,* $\Delta_{\boldsymbol{a}} \overset{def}{=} \mu^* - \mu_{\boldsymbol{a}}$*, and* $\Delta_{min} \overset{def}{=}$

$\min_{\boldsymbol{a} \in \mathcal{N}_{\boldsymbol{\pi}^*}(\boldsymbol{a}^*)} \Delta_{\boldsymbol{a}}$.

The first upper-bound (Equation (3.3)) deals with the expected number of iterations at which GRAB recommends a sub-optimal permutation while the leader is the best permutation. It derives from Theorem 2 hereafter, which detailed proof is in the appendix.

Theorem 2 (New Upper-Bound on the Regret of KL-CombUCB)**.** *We consider a combinatorial semi-bandit setting. Let E be a set of elements and $\mathcal{A} \subseteq \{0,1\}^E$ be a set of arms, where each arm $\boldsymbol{a}$ is a subset of E. Let us assume that the reward when drawing the arm $\boldsymbol{a} \in \mathcal{A}$ is $\sum_{e \in \boldsymbol{a}} c_e$, where for each element $e \in E$, c_e is an independent draw of a Bernoulli distribution of mean $\rho_e \in [0,1]$. Therefore, the expected reward when drawing the arm $\boldsymbol{a} \in \mathcal{A}$ is $\mu_{\boldsymbol{a}} = \sum_{e \in \boldsymbol{a}} \rho_e$.*

When facing this bandit setting, KL-CombUCB (CombUCB1 equiped with Kullback-Leibler indices, see Section 2.1.2) fulfills

$$\forall \boldsymbol{a} \in \mathcal{A} \text{ s.t. } \mu_{\boldsymbol{a}} \neq \mu^*,$$
$$\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{\boldsymbol{a}(t) = \boldsymbol{a}\}\right] \leqslant \frac{2K_{\boldsymbol{a}}^2}{\Delta_{\boldsymbol{a}}^2} \log T + \mathcal{O}\left(\log\log T\right),$$

and hence

$$\begin{aligned} R(T) &\leqslant \sum_{\boldsymbol{a} \in \mathcal{A}: \mu_{\boldsymbol{a}} \neq \mu^*} \frac{2K_{\boldsymbol{a}}^2}{\Delta_{\boldsymbol{a}}} \log T + \mathcal{O}\left(\log\log T\right) \\ &= \mathcal{O}\left(\frac{|\mathcal{A}| K_{max}^2}{\Delta_{min}} \log(T)\right), \end{aligned}$$

where $\mu^* \overset{def}{=} \max_{\boldsymbol{a} \in \mathcal{A}} \mu_{\boldsymbol{a}}$, $\Delta_{\boldsymbol{a}} \overset{def}{=} \mu^* - \mu_{\boldsymbol{a}}$, $\Delta_{min} \overset{def}{=} \min_{\boldsymbol{a} \in \mathcal{A}: \Delta_{\boldsymbol{a}} > 0} \Delta_{\boldsymbol{a}}$, $K_{\boldsymbol{a}} \overset{def}{=} \min_{\boldsymbol{a}^* \in \mathcal{A}: \mu_{\boldsymbol{a}^*} = \mu^*} |\boldsymbol{a} \setminus \boldsymbol{a}^*|$ is the smallest number of elements to remove from $\boldsymbol{a}$ to get an optimal arm, and $K_{max} \overset{def}{=} \max_{\boldsymbol{a} \in \mathcal{A}: \mu_{\boldsymbol{a}} \neq \mu^*} K_{\boldsymbol{a}}$.

Secondly, the expected number of iterations at which the leader is not the optimal arm (Equation (3.4)) is controlled by Lemma 2, which detailed proof is in the appendix.

Lemma 2 (Upper-Bound on the Number of Iterations of GRAB for which $\tilde{\boldsymbol{a}}(t) \neq \boldsymbol{a}^*$)**.** *Under the hypotheses of Theorem 1 and using its notations,*

$$\forall \tilde{\boldsymbol{a}} \in \mathcal{P}_K^L \setminus \{\boldsymbol{a}^*\}, \ \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{\tilde{\boldsymbol{a}}(t) = \tilde{\boldsymbol{a}}\}\right] = \mathcal{O}\left(\log\log T\right).$$

Finally, the number of iterations at which the inferred ranking on the positions is inappropriate (Equation (3.5)) is controlled by Lemma 3, which detailed proof is in the appendix.

Lemma 3 (Upper-Bound on the Number of Iterations of GRAB for which $\boldsymbol{\pi}(t) \notin \Pi_{\boldsymbol{\rho}}(\tilde{\boldsymbol{a}})$)**.** *Under the hypothesises of Theorem 1 and using its notations,*

$$\forall \tilde{\boldsymbol{a}} \in \mathcal{P}_K^L,\ \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\left\{\tilde{\boldsymbol{a}}(t) = \tilde{\boldsymbol{a}}, \tilde{\boldsymbol{\pi}}(t) \notin \Pi_{\boldsymbol{\rho}}\left(\tilde{\boldsymbol{a}}\right)\right\}\right] = \mathcal{O}\left(1\right).$$

We assemble these results to get the proof of Theorem 1.

Proof of Theorem 1. First note that, since there is a unique optimal permutation, there is a unique appropriate ranking $\boldsymbol{\pi}^*$ of positions w.r.t. $\boldsymbol{a}^*$: $\Pi_{\boldsymbol{\rho}}(\boldsymbol{a}^*) = \{\boldsymbol{\pi}^*\}$. Then, the proof is based on the following decomposition of the set $[T]$ of iterations:

$$\begin{aligned}[T] = &\bigcup_{\substack{\boldsymbol{a} \in \{\boldsymbol{a}^*\} \\ \cup \mathcal{N}_{\boldsymbol{\pi}^*}(\boldsymbol{a}^*)}} \left\{t \in [T] : \tilde{\boldsymbol{a}}(t) = \boldsymbol{a}^*, \tilde{\boldsymbol{\pi}}(t) = \boldsymbol{\pi}^*, \boldsymbol{a}(t) = \boldsymbol{a}\right\} \\ &\cup \left\{t \in [T] : \tilde{\boldsymbol{a}}(t) \neq \boldsymbol{a}^*\right\} \cup \left\{t \in [T] : \tilde{\boldsymbol{\pi}}(t) \notin \Pi_{\boldsymbol{\rho}}\left(\tilde{\boldsymbol{a}}(t)\right)\right\}.\end{aligned}$$

As for any recommendation $\boldsymbol{a}$, $\Delta_{\boldsymbol{a}} \leqslant K$, this decomposition leads to the inequality $R(T) \leqslant \sum_{\boldsymbol{a} \in \mathcal{N}_{\boldsymbol{\pi}^*}(\boldsymbol{a}^*)} \Delta_{\boldsymbol{a}} A_{\boldsymbol{a}} + KB + KC$, with

$$\begin{aligned}A_{\boldsymbol{a}} &= \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\left\{\tilde{\boldsymbol{a}}(t) = \boldsymbol{a}^*, \tilde{\boldsymbol{\pi}}(t) = \boldsymbol{\pi}^*, \boldsymbol{a}(t) = \boldsymbol{a}\right\}\right], \\ B &= \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\left\{\tilde{\boldsymbol{a}}(t) \neq \boldsymbol{a}^*\right\}\right], \\ C &= \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{\tilde{\boldsymbol{\pi}}(t) \notin \Pi_{\boldsymbol{\rho}}\left(\tilde{\boldsymbol{a}}(t)\right)\}\right].\end{aligned}$$

The term $A_{\boldsymbol{a}}$ is smaller than the expected number of times the arm $\boldsymbol{a}$ is chosen by KL-CombUCB when it plays on the set of arms $\{\boldsymbol{a}^*\} \cup \mathcal{N}_{\boldsymbol{\pi}^*}\left(\boldsymbol{a}^*\right)$. As any of these arms differs with $\boldsymbol{a}^*$ at at most two positions, Theorem 2 upper-bounds $A_{\boldsymbol{a}}$ by

$$\frac{8}{\Delta_{\boldsymbol{a}}^2} \log T + \mathcal{O}\left(\log \log T\right)$$

and hence $\sum_{\boldsymbol{a}\in\mathcal{N}_{\boldsymbol{\pi}^*}(\boldsymbol{a}^*)} \Delta_{\boldsymbol{a}} A_{\boldsymbol{a}} = \mathcal{O}\left(L/\Delta_{min}\log T\right)$ as $|\mathcal{N}_{\boldsymbol{\pi}^*}\left(\boldsymbol{a}^*\right)| = L-1$.

Note that Theorem 5 of [40], upper-bounding the regret of CombUCB1, leads to a $\mathcal{O}\left(LK/\Delta\log T\right)$ bound [2] for $\sum_{\boldsymbol{a}\in\mathcal{N}_{\boldsymbol{\pi}^*}(\boldsymbol{a}^*)} \Delta_{\boldsymbol{a}} A_{\boldsymbol{a}}$, which we reduce by a factor K by using Theorem 2.

From Lemma 2, the term B is upper-bounded by

$$B = \sum_{\tilde{\boldsymbol{a}}\in\mathcal{P}_K^L\setminus\{\boldsymbol{a}^*\}} \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\left\{\tilde{\boldsymbol{a}}(t)=\tilde{\boldsymbol{a}}\right\}\right] = \mathcal{O}\left(\log\log T\right),$$

and we upper-bound the term C with Lemma 3:

$$C = \sum_{\tilde{\boldsymbol{a}}\in\mathcal{P}_K^L} \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{\tilde{\boldsymbol{a}}(t)=\tilde{\boldsymbol{a}}, \tilde{\boldsymbol{\pi}}(t)\notin\Pi_{\boldsymbol{\rho}}\left(\tilde{\boldsymbol{a}}\right)\}\right] = \mathcal{O}\left(1\right).$$

Finally, the regret of GRAB is upper-bounded by summing these three terms, which concludes the proof.

□

3.3.1 Discussion

KL-CombUCB adapted to the PBM setting, which is presented in Section 2.1.2 has a close relationship with GRAB:

— both algorithms solve a linear sum assignment problem, they only differ from the metric to optimize: $\sum_{k=1}^{K}\hat{\rho}_{a_k,k}(t)$ for GRAB vs. $\sum_{k=1}^{K} b_{a_k,k}(t)$ for KL-CombUCB;

— both algorithms recommend the best permutation $\boldsymbol{a}$ regarding $\sum_{k=1}^{K} b_{a_k,k}(t)$, they only differ from the considered set of permutations: $\{\tilde{\boldsymbol{a}}(t)\}\cup\mathcal{N}_{\tilde{\boldsymbol{\pi}}(t)}\left(\tilde{\boldsymbol{a}}(t)\right)$ for GRAB vs. $\mathcal{P}_K^L$ for KL-CombUCB.

By considering a larger set of permutations, KL-ComUCB1 suffers a $\mathcal{O}(LK^2/\Delta_{min}\log T)$ regret (by applying [40] bound), which is higher than the upper-bound on the regret of GRAB by a factor K^2.

Assumming $\theta_1 \geqslant \cdots \geqslant \theta_L$ and $\kappa_1 \geqslant \cdots \geqslant \kappa_K$, the detailed formula for the regret upper-bound (3.6) is $\sum_{k=1}^{K-1}\frac{8\log T}{(\kappa_k-\kappa_{k+1})(\theta_k-\theta_{k+1})} + \sum_{k=K+1}^{L}\frac{8\log T}{\kappa_K(\theta_K-\theta_k)}$, where the first sum corresponds to the set of neighbors of $\boldsymbol{a}^*$ which recommend the same items as $\boldsymbol{a}^*$, and the

2. In this setting, the *ground set* is $E \overset{def}{=} \bigcup_{k\in[K]}\{(a_{\max(k-1,1)},k),(a_k,k),(a_{\min(k+1,K)},k)\} \cup \bigcup_{k\in[L]\setminus[K]}\{(a_k,K)\}$ and is of size $L+2K-2$, and any arm is composed of exactly K elements in E.

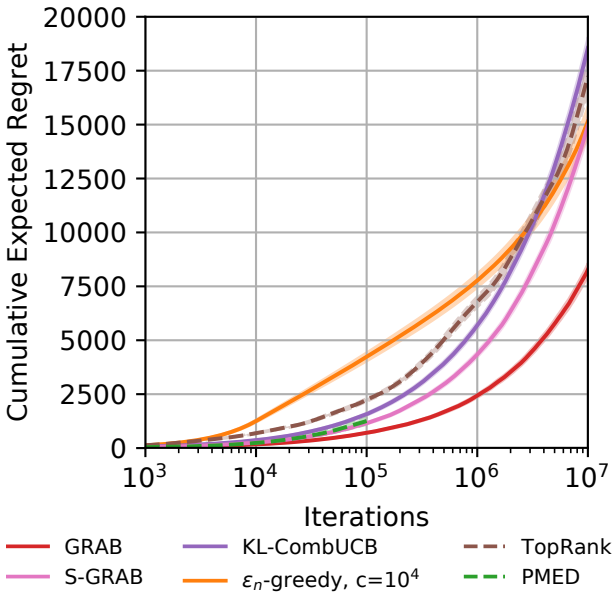
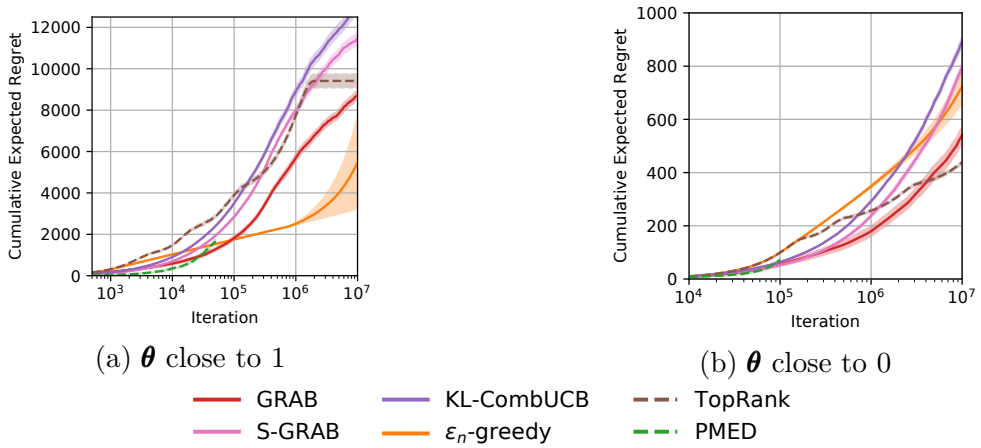


Figure 3.2: Cumulative regret w.r.t. iterations on Yandex dataset. The plotted curves correspond to the average over 200 independent sequences of recommendations (20 sequences per query). The shaded area depicts the standard error of our regret estimates.

second sum relates to the set of neighbors of $\boldsymbol{a}^*$ which replace the 'last' item in $\boldsymbol{a}^*$. Hence, the number of displayed items does not impact the total number of terms, but the gaps $\Delta_{\boldsymbol{a}}$.

Note also that GRAB is, by design, robust to miss-specifications. Typically, GRAB would properly handle a matrix $\boldsymbol{\rho} = \boldsymbol{\theta}^T \boldsymbol{\kappa} + \mathcal{E}$, if $\max_{i,j} |\mathcal{E}_{i,j}|$ is smaller than half of the minimum gap between two entries of the matrix $\boldsymbol{\theta}^T \boldsymbol{\kappa}$.

However, if there is a set of optimal recommendations $\mathcal{A}^*$ (instead of a unique one), after convergence, the leader will be picked in that set at each iteration. So the neighborhood of each optimal recommendation will be explored, and we will get a regret bound in $\mathcal{O}(|\mathcal{A}^*|L)$. This behavior questions the applicability of unimodality to the *Cascading Model* (CM), as with this model there is at least $K!$ optimal recommendations. Moreover, while Assumption 1 is valid for CM and the *Dependent Click Model* (DCM), our setting also assumes the existence of the matrix $\boldsymbol{\rho}$, which is false for CM and DCM: in both settings the probability of clicking on item i in position ℓ depends on other displayed items.



(a) $\boldsymbol{\theta}$ close to 1

(b) $\boldsymbol{\theta}$ close to 0

Figure 3.3: Cumulative regret w.r.t. iterations on simulated data. The plotted curves correspond to the average over 20 independent sequences of recommendations. The shaded area depicts the standard error of our regret estimates. For ϵ_n-Greedy, c is set to 10^5 when $\boldsymbol{\theta}$ is close to 0, and to 10^3 when $\boldsymbol{\theta}$ is close to 1.

3.4 Practical results

In this section, we compare GRAB to PMED [37], to TopRank [43], to ϵ_n-Greedy, to Static Graph for unimodal RAnking Bandit (S-GRAB), a simplified version of GRAB presented in Appendix A.7, and to KL-CombUCB, an adaptation of CombUCB1 [40]. The experiments are conducted on the Yandex dataset (see Section 1.3.4) and on purely simulated data (see Section 1.3.4). We use the cumulative regret to evaluate the performance of each algorithm, where the cumulative regret is averaged over 20 independent runs of $T = 10^7$ iterations each. Code and data for replicating our experiments are available at `https://github.com/gaudel/ranking_bandits`.

We mostly compare GRAB to algorithms presented in Chapter2 but also to ϵ_n-Greedy, which is a simple yet efficient algorithm for standart usecases. At each time-stamp t, ϵ_n-Greedy computes an estimation $(\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\kappa}})$ of parameters $(\boldsymbol{\theta}, \boldsymbol{\kappa})$ by applying Singular Value Decomposition (SVD) to the collected data. Let us denote $\hat{\boldsymbol{a}}(t)$ the recommendation with the highest expected reward given the inferred values $(\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\kappa}})$. A greedy algorithm would recommend $\hat{\boldsymbol{a}}(t)$. Since this algorithm never explores, it may end-up recommending a sub-optimal affectation. ϵ_n-Greedy counters this by randomly replacing each item of the recommendation with a probability $\varepsilon(t) = c/t$, where c is a hyper-parameter to be tuned. In the following, we plot the results obtained with the best possible value for c, while trying c in $\{10^0, 10^1, \ldots, 10^6\}$. Note that the best value for c varies from a dataset to another.

In the following, the tunning of c leads to use $c = 10^4$ for Yandex dataset, $c = 10^5$ for simulated dataset with $\boldsymbol{\theta}$ close to 0 and $c = 10^3$ for simulated dataset with $\boldsymbol{\theta}$ close to 1.

Figure 3.2 shows the results for the algorithms on Yandex and Figure 3.3 on the simulated data. We measure the performance of each algorithm according to the cumulative regret (see Equation 1.6). It is the sum, over T consecutive recommendations, of the difference between the expected reward of the best answer and of the answer of a given recommender system. The best algorithm is the one with the lowest regret. We average the results of each algorithm over 20 independent sequences of recommendations per query or simulated setting. Although PMED theoretically yields an asymptotically optimal regret, we stop it at iteration $t = 10^5$ due to its heavy computation-time.

Ablation Study The two main ingredients of GRAB are the use of a graph to explore the set of recommendations, and the online inference of this graph. Without these ingredients, GRAB boils down to KL-CombUCB which recommends at each iteration the best permutation given the sum of indices $b_{i,k}$ and has a $\mathcal{O}\left(LK^2/\Delta \log T\right)$ regret. With only the first ingredient (namely a static graph of degree $\Theta(LK)$), we get S-GRAB which regret is upper-bounded by $\mathcal{O}\left(LK/\Delta \log T\right)$, while GRAB's regret is upper-bounded by $\mathcal{O}\left(L/\Delta \log T\right)$ thanks to a set of graphs of degree $L-1$.

We want to assert the empirical impact of these ingredients. On Figures 3.2 and 3.3, we see that GRAB has a better regret than S-GRAB and KL-CombUCB in every settings. This confirms that the proposed graphs are relevant to explore the set of recommendations, and that GRAB quickly infer the appropriate graph in the family of potential ones.

Results Analysis Figure 3.2 compares the empirical regret of all algorithms on Yandex dataset. GRAB is the best with a regret at $T = 10^7$ about two time smaller than the rest of the algorithms.

Figure 3.3 shows our results on purely simulated data illustrating extreme settings, where values of $\boldsymbol{\theta}$ are extremely close to 0 (Figure 3.3b) or close to 1 (Figure 3.3a) even though these settings are less realistic. In both settings, GRAB is in the top-2 algorithms. However, while TopRank provides better or similar result as GRAB at iteration 10^7, its regret is higher than the one of GRAB up to iteration $t = 4 \times 10^6$. TopRank only catches-up GRAB at the end of the sequences of recommendations. We note that in the setting close to 1, TopRank manages to find the perfect order after 10^6 iterations. In this setting too, ε_n-greedy has better performance during the 10^6 first iterations, but suffers from its

Table 3.2: Average computation time for sequences of 10^7 recommendations vs. all queries of Yandex dataset

Algorithm	**(hour/min)**	**trial (ms)**
GRAB	2h24	0.9
S-GRAB	9h56	3.6
ε_n-greedy $c = 10^4$	1h13	0.4
KL-CombUCB	2h03	0.7
PMED	474h13*	170
TopRank	9h29	3

* Extrapolation from 10^5 recommendations.

greedy behaviour during the last steps with a large variance.

Computation Time As shown in Table 3.2, the fastest algorithm is ε_n-greedy. KL-CombUCB and GRAB are two times slower. The exploration of S-GRAB multiplies its computation time by 4 compared to GRAB. TopRank is about three times slower than GRAB.

3.5 Conclusion

We saw that unimodal bandit is a promising way to tackle list recommendations under the Position Based Model as this model leads to an easy-to-use metric to find the best recommendation w.r.t. $b_{a_k,k}(t)$. This method can be put closer to UCB method. Other efficient Bandits methods are based on Thompson Sampling. In the context of PBM setting, the application of Thompson Sampling requires drawing samples after an unusual law. We'll see in the next chapter that methods exist to simulate those draw.

Chapter 4

MCMC bandits for PBM

Contents

4.1 Thompson sampling with approximation approaches 70

4.1.1 Approximation based on Metropolis Hasting 71

4.1.2 Approximation based on Langevin gradient descent 76

4.1.3 Overall complexity 78

4.2 Practical results 78

4.3 Conclusion 89

Markov chain Monte Carlo (MCMC) is a class of sampling methods which solve the problem evoked at the end of the precedent chapter: How to use Thompson Sampling Bandit algorithm on the exact posterior law induced by PBM. Two of these sampling methods are used in this work: Metropolis Hasting and Langevin gradient descent.Their adaptation in our bandit setting is explained in this chapter. A preliminary version of these contribution has been presented at IDA'21 [23] and the final version is under review in ACM Transactions on Information Systems *journal.*

The proposed approaches handle a position-based online learning to rank (PB-OLR) bandit setting, defined in Section 1.3 and covering PBM click model. This chapter introduces a family of bandit algorithms designed to handle PBM with a semi-bandits Thompson sampling framework. These algorithms do not require the knowledge of the probability of a user to look at a given position: they learn this probability from past recommendations/feedbacks. This is a strong improvement w.r.t. previous attempts in this research line [36, 41] as it allows the use of our algorithms in contexts where this information is not obvious: a web-page with a layout which often changes and, with it, the probability of a user to look at a given position. Besides, even with stable layouts, it is easier to apply a framework which learns the attractiveness of both items and positions than having two separate modules: an online learning approach dedicated to the positions and a bandit algorithm dedicated to the items. This improvement results from the use of Markov Chain

Monte Carlo (MCMC) [50, 18] methods to sample parameters given an approximation of their posterior *distribution which monotonically approaches the posterior distribution. While MCMC methods are well-known and extensively used in Bayesian statistics, they were rarely used for Thomson Sampling [35, 17, 56, 48] and it is the first time that the Metropolis-Hastings framework is used in the PBM setting.*

As mentioned in Section 2.1, previous work [36, 41] apply Thompson sampling to tackle PBM but limit themselves to a setting where $\boldsymbol{\kappa}$ is known. The distributions rising from this assumption is easier than the one which arises from $\boldsymbol{\kappa}$ being unknown. In the following, we propose to use Metropolis-Hastings framework and Langevin gradient descent, two strategies to approximate draw from various distributions, to handle this harder distribution.

Other works [17, 56, 35, 48] investigate a large range of distribution approximation strategies to apply TS framework to the distributions arising from various setting such as the contextual bandit *setting. Overall, these articles handle a pure bandit setting while we are in a semi-bandits setting: for each recommendation we receive as reward a list of 1 or 0 (click or not). As most of commercial website can track precisely on which product each client clicks, we aim at exploiting that fine-grain information.*

This chapter is organised as follow: Section 4.1 presents the different approximation approaches uses to build PB-MHB and PB-LB. Empirical performances are presented Section 4.2. We conclude in Section 4.3

4.1 Thompson sampling with approximation approaches

We handle the setting presented in Section 1.3 with the online recommender system depicted by Algorithm 7. We present here two versions of this method, which is based on the Thompson sampling framework [62, 1] and use standard statistical methods to approximate the parameters of the reward law. Thus, we firstly look at rewards with a fully Bayesian point of view: we assume that they follow the statistical model depicted in Section 1.3, and we choose a uniform prior on the parameters $\boldsymbol{\theta}$ and $\boldsymbol{\kappa}$. Therefore the posterior probability for these parameters given the previous observations $D(t)$ is

$$\mathbb{P}\left(\boldsymbol{\theta}, \boldsymbol{\kappa} | D(t)\right) \propto \prod_{i=1}^{L} \prod_{k=1}^{K} \left(\theta_i \kappa_k\right)^{S_{i,k}(t)} \left(1 - \theta_i \kappa_k\right)^{F_{i,k}(t)}, \tag{4.1}$$

where $S_{i,k}(t) = \sum_{s=1}^{t-1} \mathbb{1}_{i_k(s)=i} \mathbb{1}_{r_k(s)=1}$ denotes the number of times the item i has been clicked while being displayed in position k from iteration 1 to $t-1$, and $F_{i,k}(t) =$

Algorithm 7 Thompson Sampling with MCMC approximations

$D(1) \leftarrow \{\}$
for $t = 1, \dots$ **do**
  draw $(\tilde{\boldsymbol{\theta}}, \tilde{\boldsymbol{\kappa}}) \sim \tilde{\mathbb{P}}\left(\boldsymbol{\theta}, \boldsymbol{\kappa} | D(t)\right)$ using MCMC approximation (see Algorithm 8 or 9)
  display the K items with greatest value in $\tilde{\boldsymbol{\theta}}$, ordered by decreasing values of $\tilde{\boldsymbol{\kappa}}$
  get rewards $\boldsymbol{r}(t)$
  $D(t+1) \leftarrow D(t) \cup (\boldsymbol{a}(t), \boldsymbol{r}(t))$
end for

$\sum_{s=1}^{t-1} \mathbb{1}_{i_k(s)=i} \mathbb{1}_{r_k(s)=0}$ denotes the number of times the item i has not been clicked while being displayed in position k from iteration 1 to $t-1$.

Secondly, we choose the recommendation $\boldsymbol{a}(t)$ at iteration t according to its posterior probability of being the best arm. To do so, we draw a sample $(\tilde{\boldsymbol{\theta}}, \tilde{\boldsymbol{\kappa}})$ of parameters $(\boldsymbol{\theta}, \boldsymbol{\kappa})$ according to their posterior distribution, we keep the best items given $\tilde{\boldsymbol{\theta}}$, and we display them in the right order given $\tilde{\boldsymbol{\kappa}}$.

The posterior distribution in Eq. (4.1) of parameters $\boldsymbol{\theta}$ and $\boldsymbol{\kappa}$ does not belong to a standard family of distributions. Hence we have to resort to approximate sampling. We propose the use of Metropolis-Hasting and Langevin algorithms, leading respectively to the PB-MHB (Position Based Metropolis-Hastings Bandit) family of algorithms and to PB-LB (Position Based Langevin gradient Bandit) algorithm that we detail in the following.

4.1.1 Approximation based on Metropolis Hasting

As already mentioned, the distribution in Eq. (4.1) does not correspond to a well-known distribution. [36, 41] circumvent this problem by considering that $\boldsymbol{\kappa}$ is known in order to manipulate L independent simpler distributions $\mathbb{P}_i\left(\theta_i | \boldsymbol{\theta}_{-i}, \boldsymbol{\kappa}, D(t)\right)$, where $\boldsymbol{\theta}_{-i}$ denotes the components of $\boldsymbol{\theta}$ except for the i-th one. These approaches gives good results but fails when $\boldsymbol{\kappa}$ is unknown. Indeed, by having $\boldsymbol{\kappa}$ and $\boldsymbol{\theta}$ both unknown, we have to handle a law for which the components $\theta_1, \dots, \theta_L$ and $\kappa_1, \dots, \kappa_K$ are correlated (see Equation 4.1). We propose the PB-MHB family of algorithms to address this issue. This family handles the unusual distribution given in Eq. (4.1) thanks to a carefully designed Metropolis-Hastings algorithm [50] (cf. Algorithm 8). This algorithm consists in building a sequence of m samples $(\boldsymbol{\theta}^{(1)}, \boldsymbol{\kappa}^{(1)}), \dots, (\boldsymbol{\theta}^{(m)}, \boldsymbol{\kappa}^{(m)})$ such that $(\boldsymbol{\theta}^{(m)}, \boldsymbol{\kappa}^{(m)})$ follows a good approximation of the targeted distribution. It is based on a Markov chain on parameters $(\boldsymbol{\theta}, \boldsymbol{\kappa})$ which admits the targeted probability distribution as its unique stationary distribution. It start

Algorithm 8 Metropolis-Hastings applied to the distribution of Equation (4.1)

Require: $D(t)$: previous recommendations and rewards
Require: $q_1^\theta, \dots, q_L^\theta$ and $q_1^\kappa, \dots, q_K^\kappa$: densities of proposal for each components of $(\boldsymbol{\theta}, \boldsymbol{\kappa})$
Require: m: number of steps
Require: (if Gaussian Random Walk proposal) $\sigma = c/\sqrt{t}$: width of Gaussian random-walk steps

1: $(\boldsymbol{\theta}^{(0)}, \boldsymbol{\kappa}^{(0)}) \leftarrow \begin{cases} \text{draw } (\boldsymbol{\theta}^{(0)}, \boldsymbol{\kappa}^{(0)}) \text{ after uniform distribution} & \text{, if random start} \\ \text{reuse } (\tilde{\boldsymbol{\theta}}, \tilde{\boldsymbol{\kappa}}) \text{ from previous recommendation} & \text{, otherwise} \end{cases}$
2: $\kappa_1^{(0)} \leftarrow 1$
3: **for** $s = 1, \dots, m$ **do**
4: $\quad (\dot{\boldsymbol{\theta}}, \dot{\boldsymbol{\kappa}}) \leftarrow (\boldsymbol{\theta}^{(s-1)}, \boldsymbol{\kappa}^{(s-1)})$
5: $\quad$ **for** $i = 1, \dots, L$ **do**
6: $\quad\quad$ draw $\tilde{\theta}_i$ from the proposal law of density $q_i^\theta\left(\tilde{\theta}_i \mid \dot{\theta}_i, \dot{\boldsymbol{\theta}}_{-i}, \dot{\boldsymbol{\kappa}}, D(t)\right)$
7: $\quad\quad \dot{\theta}_i \leftarrow \tilde{\theta}_i$ **with prob.** $p_{acc}^\theta \overset{def}{=} \min\left(1, \frac{\mathbb{P}_i\left(\tilde{\theta}_i | \dot{\boldsymbol{\theta}}_{-i}, \dot{\boldsymbol{\kappa}}, D(t)\right)}{\mathbb{P}_i\left(\dot{\theta}_i | \dot{\boldsymbol{\theta}}_{-i}, \dot{\boldsymbol{\kappa}}, D(t)\right)} \frac{q_i^\theta\left(\dot{\theta}_i | \tilde{\theta}_i, \dot{\boldsymbol{\theta}}_{-i}, \dot{\boldsymbol{\kappa}}, D(t)\right)}{q_i^\theta\left(\tilde{\theta}_i | \dot{\theta}_i, \dot{\boldsymbol{\theta}}_{-i}, \dot{\boldsymbol{\kappa}}, D(t)\right)}\right)$
8: $\quad$ **end for**
9: $\quad$ **for** $k = 2, \dots, K$ **do**
10: $\quad\quad$ draw $\tilde{\kappa}_i$ from the proposal law of density $q_k^\kappa\left(\tilde{\kappa}_i \mid \dot{\kappa}_k, \dot{\boldsymbol{\kappa}}_{-k}, \dot{\boldsymbol{\theta}}, D(t)\right)$
11: $\quad\quad \dot{\kappa}_k \leftarrow \tilde{\kappa}_k$ **with prob.** $p_{acc}^\kappa \overset{def}{=} \min\left(1, \frac{\mathbb{P}_k\left(\tilde{\kappa}_i | \dot{\boldsymbol{\kappa}}_{-k}, \dot{\boldsymbol{\theta}}, D(t)\right)}{\mathbb{P}_k\left(\dot{\kappa}_k | \dot{\boldsymbol{\kappa}}_{-k}, \dot{\boldsymbol{\theta}}, D(t)\right)} \frac{q_k^\kappa\left(\dot{\kappa}_k | \tilde{\kappa}_i, \dot{\boldsymbol{\kappa}}_{-k}, \dot{\boldsymbol{\theta}}, D(t)\right)}{q_k^\kappa\left(\tilde{\kappa}_i | \dot{\kappa}_k, \dot{\boldsymbol{\kappa}}_{-k}, \dot{\boldsymbol{\theta}}, D(t)\right)}\right)$
12: $\quad$ **end for**
13: $\quad (\boldsymbol{\theta}^{(s)}, \boldsymbol{\kappa}^{(s)}) \leftarrow (\dot{\boldsymbol{\theta}}, \dot{\boldsymbol{\kappa}})$
14: **end for**
15:
16: **return** $(\boldsymbol{\theta}^{(m)}, \boldsymbol{\kappa}^{(m)})$

from an initial pair $(\boldsymbol{\theta}^{(0)}, \boldsymbol{\kappa}^{(0)})$ which can be drawn from a uniform distribution or reused from previous recommendation.

At step s, the sample $(\boldsymbol{\theta}^{(s)}, \boldsymbol{\kappa}^{(s)})$ moves toward sample $(\boldsymbol{\theta}^{(s+1)}, \boldsymbol{\kappa}^{(s+1)})$ by applying $(L + K - 1)$ transitions: one per item and one per position except for κ_1. Let us start by focusing on the transition regarding item i (Lines 6–7) and denote $(\dot{\boldsymbol{\theta}}, \dot{\boldsymbol{\kappa}})$ the sample before the transition.

The algorithm aims at sampling a new value for $\dot{\theta}_i$ according to its posterior probability given other parameters and the previous observations $D(t)$:

$$\mathbb{P}_i\left(\dot{\theta}_i | \dot{\boldsymbol{\theta}}_{-i}, \dot{\boldsymbol{\kappa}}, D(t)\right) \propto \prod_{k=1}^{K} \dot{\theta}_i^{S_{i,k}(t)} \left(1 - \dot{\theta}_i \dot{\kappa}_k\right)^{F_{i,k}(t)}. \tag{4.2}$$

This transition consists in two steps:

1. draw a *candidate* value $\tilde{\theta}$ after a *proposal* probability distribution $q_i^\theta\left(\tilde{\theta}_i \mid \dot{\theta}_i, \dot{\boldsymbol{\theta}}_{-i}, \dot{\boldsymbol{\kappa}}, D(t)\right)$ to be discussed later on;

2. *accept* that candidate or keep the previous sample:

$$\dot{\theta}_i \leftarrow \begin{cases} \tilde{\theta}_i & \text{, with prob. } p_{acc}^\theta \\ \dot{\theta}_i & \text{, otherwise} \end{cases},$$

with $p_{acc}^\theta \overset{def}{=} \min\left(1, \frac{\mathbb{P}_i\left(\tilde{\theta}_i|\dot{\boldsymbol{\theta}}_{-i},\dot{\boldsymbol{\kappa}},D(t)\right)}{\mathbb{P}_i\left(\dot{\theta}_i|\dot{\boldsymbol{\theta}}_{-i},\dot{\boldsymbol{\kappa}},D(t)\right)} \frac{q_i^\theta\left(\dot{\theta}_i|\tilde{\theta}_i,\dot{\boldsymbol{\theta}}_{-i},\dot{\boldsymbol{\kappa}},D(t)\right)}{q_i^\theta\left(\tilde{\theta}_i|\dot{\theta}_i,\dot{\boldsymbol{\theta}}_{-i},\dot{\boldsymbol{\kappa}},D(t)\right)}\right).$

This acceptance step yields two behaviours:
— $\frac{\mathbb{P}_i\left(\tilde{\theta}_i|\dot{\boldsymbol{\theta}}_{-i},\dot{\boldsymbol{\kappa}},D(t)\right)}{\mathbb{P}_i\left(\dot{\theta}_i|\dot{\boldsymbol{\theta}}_{-i},\dot{\boldsymbol{\kappa}},D(t)\right)}$ measures how likely the candidate value is compared to the previous one, w.r.t. the posterior distribution,
— $\frac{q_i^\theta\left(\dot{\theta}_i|\tilde{\theta}_i,\dot{\boldsymbol{\theta}}_{-i},\dot{\boldsymbol{\kappa}},D(t)\right)}{q_i^\theta\left(\tilde{\theta}_i|\dot{\theta}_i,\dot{\boldsymbol{\theta}}_{-i},\dot{\boldsymbol{\kappa}},D(t)\right)}$ prevents preferring candidates easily reached by the proposal q_i^θ.

The sampling process for the parameter $\dot{\kappa}_k$ is similar (Lines 10–11). The Metropolis-Hastings step is based on the proposal $q_k^\kappa\left(\tilde{\kappa}_i \mid \dot{\kappa}_k, \dot{\boldsymbol{\kappa}}_{-k}, \dot{\boldsymbol{\theta}}, D(t)\right)$ and aims at the probability

$$\mathbb{P}_k\left(\dot{\kappa}_k|\dot{\boldsymbol{\theta}}, \dot{\boldsymbol{\kappa}}_{-k}, D(t)\right) \propto \prod_{i=1}^{L} \dot{\kappa}_k^{S_{i,k}(t)} \left(1 - \dot{\theta}_i\dot{\kappa}_k\right)^{F_{i,k}(t)}. \tag{4.3}$$

The proposal laws are hyper-parameters of Algorithm 8. This flexibility allows to use generic proposals as well as task-specific ones. We implement and detail in the following four different laws: a Truncated Gaussian Random Walk (TGRW), a Logit Gaussian Random Walk (LGRW), a proposal using the approximated law used in [36] named here *PseudoView* and the one from [41] named *MaxPos*. The first two are generic and can be applied to any situation. The later two were borrowed from previous works tackling a similar task.

Truncated Gaussian Random Walk This proposal draws $\tilde{\theta}$ (respectively $\tilde{\kappa}$) from $\mathcal{N}\left(\dot{\theta}_i, \sigma\right)$ (resp. $\mathcal{N}\left(\dot{\kappa}_k, \sigma\right)$) with a Gaussian step of *standard deviation* σ. As our targeted law is bounded in $[0, 1]$, we truncate this distribution by rejecting candidate values out of $[0, 1]$. Note that due to the truncation, the probability to get the proposal $\tilde{\theta}_i$ starting from $\dot{\theta}_i$ is

$$q_i^\theta\left(\tilde{\theta}_i \mid \dot{\theta}_i, \dot{\boldsymbol{\theta}}_{-i}, \dot{\boldsymbol{\kappa}}, D(t)\right) = \frac{\phi(\tilde{\theta}_i \mid \dot{\theta}_i, \sigma)}{\Delta\Phi_\sigma(\dot{\theta}_i)},$$

where $\phi(\cdot \mid \dot{\theta}_i, \sigma)$ is the probability associated to the Gaussian distribution with mean $\dot{\theta}_i$ and standard deviation σ, $\Phi(\cdot \mid \dot{\theta}_i, \sigma)$ is its cumulative distribution function, and

$$\Delta\Phi_\sigma(\dot{\theta}_i) = \Phi(1 \mid \dot{\theta}_i, \sigma) - \Phi(0 \mid \dot{\theta}_i, \sigma).$$

The probability to get the proposal $\dot{\theta}_i$ starting from $\tilde{\theta}_i$ is similar, which reduces the ratio of proposal probabilities at Line 7 to

$$\frac{q_i^\theta\left(\dot{\theta}_i \mid \tilde{\theta}_i, \dot{\boldsymbol{\theta}}_{-i}, \dot{\boldsymbol{\kappa}}, D(t)\right)}{q_i^\theta\left(\tilde{\theta}_i \mid \dot{\theta}_i, \dot{\boldsymbol{\theta}}_{-i}, \dot{\boldsymbol{\kappa}}, D(t)\right)} = \frac{\Delta\Phi_\sigma\left(\dot{\theta}_i\right)}{\Delta\Phi_\sigma\left(\tilde{\theta}_i\right)}.$$

As we will see in section 4.2, this proposal is efficient while being generic.

Logit Gaussian Random Walk The second set of proposals avoid the truncation by applying a Gaussian random walk on the logit of each parameter θ_i and κ_k. As an example, the proposal for item i is drawn as

$$\tilde{\theta}_i = \text{logit}^{-1}\left(\hat{\theta}_i^\ell\right), \tag{4.4}$$

where $\hat{\theta}_i^\ell \sim \mathcal{N}\left(\text{logit}\left(\dot{\theta}_i\right), \sigma\right)$, and $\text{logit}(p) = \log\left(\frac{p}{1-p}\right)$, for $p \in (0,1)$. Note that the inverse of the logit function is the logistic function i.e. $\text{logit}^{-1}(x) = \frac{1}{1+\exp(-x)}$, with $x \in \mathbb{R}$, which enforces $\tilde{\theta}_i \in [0,1]$.

As the proposal is mainly a Gaussian random walk, we have:

$$\frac{q_i^\theta\left(\dot{\theta}_i \mid \tilde{\theta}_i, \dot{\boldsymbol{\theta}}_{-i}, \dot{\boldsymbol{\kappa}}, D(t)\right)}{q_i^\theta\left(\tilde{\theta}_i \mid \dot{\theta}_i, \dot{\boldsymbol{\theta}}_{-i}, \dot{\boldsymbol{\kappa}}, D(t)\right)} = \frac{\phi\left(\text{logit}\left(\dot{\theta}_i\right) \mid \text{logit}\left(\tilde{\theta}_i\right), \sigma\right)}{\phi\left(\text{logit}\left(\tilde{\theta}_i\right) \mid \text{logit}\left(\dot{\theta}_i\right), \sigma\right)} = 1.$$

The transition regarding the parameter κ_k involves the same framework, and the corresponding set of proposals gives an alternative to TGRW for unspecific tasks.

PseudoView We also design a set of proposals by adapting the posterior law used in [36], which assumes that the $\boldsymbol{\kappa}$ parameter is known. This proposal is specific to our task which should accelerate the convergence of the Metropolis-Hastings method.

For the parameter θ_i, the proposal is defined as follows. We define $\tilde{N}_i(t) = \sum_{k=1}^K \dot{\kappa}_k \sum_{s=1}^{t-1} \mathbb{1}_{i_k(s)=i}$, the pseudo-expected number of times that item i has been observed. Note that this pseudo-expectation depends on the current estimate $\dot{\boldsymbol{\kappa}}$ of $\boldsymbol{\kappa}$. We draw the next candidate $\tilde{\theta}_i$ as

follow:

$$\tilde{\theta}_i \sim \text{Beta}\left(S_i(t)+1, \max\left(\tilde{N}_i(t)-S_i(t)+1, 1\right)\right), \tag{4.5}$$

with $S_i(t) = \sum_{k=1}^{K} S_{i,k}(t)$ the sum of the clicks obtained by item i over all the positions until iteration $t-1$, and Beta the *beta distribution*.

Thus, the ratio of the proposal probabilities becomes

$$\frac{q_i^\theta\left(\dot{\theta}_i \mid \tilde{\theta}_i, \dot{\boldsymbol{\theta}}_{-i}, \dot{\boldsymbol{\kappa}}, D(t)\right)}{q_i^\theta\left(\tilde{\theta}_i \mid \dot{\theta}_i, \dot{\boldsymbol{\theta}}_{-i}, \dot{\boldsymbol{\kappa}}, D(t)\right)} = \frac{\dot{\theta}_i^{S_i(t)}(1-\dot{\theta}_i)^{\tilde{N}_i(t)-S_i(t)}}{\tilde{\theta}_i^{S_i(t)}(1-\tilde{\theta}_i)^{\tilde{N}_i(t)-S_i(t)}}.$$

The transition regarding the parameter κ_k involves the same framework, with $S_k(t) = \sum_{i=1}^{L} S_{i,k}(t)$ the sum of the clicks obtained at position k over all items placed there until iteration $t-1$, and $\tilde{N}_k(t) = \sum_{i=1}^{L} \dot{\theta}_i^{(t)} \sum_{s=1}^{t-1} \mathbb{1}_{i_k(s)=i}$ the pseudo-expected number of times position k has been clicked, i.e. the proportion of users to click at this position regardless of the item.

This set of proposals is specifically designed for our problem. It is based on the estimation of the average impact of each item, independently of the position it as been displayed at, and on the estimation of the average impact of each position. Nevertheless, we will show in our experiments that this *PseudoView* proposal is less efficient than the generic ones, such as the previously described proposals Truncated Gaussian Random Walk and Logit Gaussian Random Walk.

MaxPos Finally, we introduce the *MaxPos* proposal from [41]. As for *PseudoView*, this proposal is specific to our task. For each item i, we identify the position k_{max} in which it has been displayed the most, and we restrict ourselves to the statistics $S_{i,k_{max}}$ and $F_{i,k_{max}}$ gathered at this position to draw a candidate $\tilde{\theta}_i$. As drawing from $\text{Beta}(S_{i,k_{max}}(t)+1, F_{i,k_{max}}(t)+1)$ leads to an estimation of the product $\theta_i \kappa_{k_{max}}$ we define our proposal as

$$\tilde{\theta}_i = \frac{x}{\dot{\kappa}_{k_{max}}}, \quad \text{with} \quad x \sim \text{Beta}\left(S_{i,k_{max}}(t)+1, F_{i,k_{max}}(t)+1\right), \tag{4.6}$$

where $k_{max} = \text{argmax}_{1 \leqslant k \leqslant K}(S_{i,k}(t) + F_{i,k}(t))$.

Thus, the ratio of the proposal probabilities becomes

$$\frac{q_i^\theta\left(\dot{\theta}_i \mid \tilde{\theta}_i, \dot{\boldsymbol{\theta}}_{-i}, \dot{\boldsymbol{\kappa}}, D(t)\right)}{q_i^\theta\left(\tilde{\theta}_i \mid \dot{\theta}_i, \dot{\boldsymbol{\theta}}_{-i}, \dot{\boldsymbol{\kappa}}, D(t)\right)} = \frac{(\dot{\theta}_i \dot{\kappa}_{k_{max}}^{(t)})^{S_{i,k_{max}}(t)}(1-\dot{\theta}_i \dot{\kappa}_{k_{max}}^{(t)})^{F_{i,k_{max}}(t)}}{(\tilde{\theta}_i \dot{\kappa}_{k_{max}}^{(t)})^{S_{i,k_{max}}(t)}(1-\tilde{\theta}_i \dot{\kappa}_{k_{max}}^{(t)})^{F_{i,k_{max}}(t)}}.$$

Algorithm 9 Langevin Algorithm applied to the inverse log-likelihood $\hat{U}$

Require: $(\boldsymbol{\theta}^{(0)}, \boldsymbol{\kappa}^{(0)})$: use particle $(\boldsymbol{\theta}^{(m)}, \boldsymbol{\kappa}^{(m)})$ from the previous call to Algorithm 9
Require: $D(t)$: previous recommendations and rewards
Require: m: number of steps
Require: $h = h_0/t$: gradient step-size
Require: γ: noise-parameter for the final step
  for $s = 1, \dots, m$ **do**
    Compute $\nabla \hat{U}_{\boldsymbol{\theta}} = \left[\frac{\partial \hat{U}}{\partial \theta_1}(\boldsymbol{\theta}^{(s-1)}, \boldsymbol{\kappa}^{(s-1)}), \dots, \frac{\partial \hat{U}}{\partial \theta_L}(\boldsymbol{\theta}^{(s-1)}, \boldsymbol{\kappa}^{(s-1)})\right]$
    Compute $\nabla \hat{U}_{\boldsymbol{\kappa}} = \left[\frac{\partial \hat{U}}{\partial \kappa_1}(\boldsymbol{\theta}^{(s-1)}, \boldsymbol{\kappa}^{(s-1)}), \dots, \frac{\partial \hat{U}}{\partial \kappa_K}(\boldsymbol{\theta}^{(s-1)}, \boldsymbol{\kappa}^{(s-1)})\right]$
    Sample $\left[\boldsymbol{\theta}^{(s)}, \boldsymbol{\kappa}^{(s)}\right] \sim \mathcal{N}\left(\left[\boldsymbol{\theta}^{(s-1)}, \boldsymbol{\kappa}^{(s-1)}\right] - h\left[\nabla \hat{U}_{\boldsymbol{\theta}}, \nabla \hat{U}_{\boldsymbol{\kappa}}\right],\ 2hI_{L+K}\right)$
  end for
  Sample $\left[\tilde{\boldsymbol{\theta}}, \tilde{\boldsymbol{\kappa}}\right] \sim \mathcal{N}\left(\left[\boldsymbol{\theta}^{(m)}, \boldsymbol{\kappa}^{(m)}\right], \frac{1}{t\gamma} I_{L+K}\right)$

  return $(\tilde{\boldsymbol{\theta}}, \tilde{\boldsymbol{\kappa}})$ and $(\boldsymbol{\theta}^{(m)}, \boldsymbol{\kappa}^{(m)})$

The transition regarding the parameter κ_k involves the same framework, with

$$i_{max} = \underset{1 \leqslant i \leqslant L}{\operatorname{argmax}}(S_{i,k}(t) + F_{i,k}(t)).$$

By restricting themselves to the most used positions (respectively the most displayed items), these proposals loose part of the gathered information. However, in the context of a bandit setting, each item should be displayed at the right position most of the time, therefore, the impact of this data loss should become negligible when the time tends to infinity.

Overall, PB-MHB (i) offers a way to tackle settings with unusual posterior distribution and (ii) is a flexible tool which can take different proposal as parameter to adapt to the problem we are facing. To highlight this flexibility, we have presented two generic proposals (TGRW and LGRW) and two proposals specific to our application which take inspiration from [36] and [41].

4.1.2 Approximation based on Langevin gradient descent

Other statistical methods can be coupled with the Thompson Sampling method to approximate the parameters from the exact reward law such as the Langevin Gradient Descent. This is the approach used in [48] except that the corresponding bandit setting is composed of a set of independent arms. We now present how an Langevin gradient

descent may be used in our more entangled setting:

— the arms (the recommendations) are no more independents, they share the same set of parameters $\boldsymbol{\kappa}$ and $\boldsymbol{\theta}$;
— each observation (the click/no-click on item i in position k at iteration t) results from the combination of two parameters: θ_i and κ_k;
— at each iteration, we observe K random outputs instead of only one.

We propose PB-LB (for Position Based Langevin gradient Bandit) which consists in using Algorithm 9 to sample $(\tilde{\boldsymbol{\theta}}, \tilde{\boldsymbol{\kappa}})$ at Line 3 of Algorithm 7. Algorithm 9 applies an Langevin strategy to the function

$$\hat{U}(\boldsymbol{\theta}, \boldsymbol{\kappa}) = -\log \mathbb{P}\left(\boldsymbol{\theta}, \boldsymbol{\kappa} | D(t)\right),$$

which is the opposite of the log-likelihood of parameters $\boldsymbol{\theta}$ and $\boldsymbol{\kappa}$, assuming uniform prior for these parameters.

At each iteration t, an Langevin algorithm is run on m steps, and each step consists in updating the parameters given the gradient of $\hat{U}$ and adding Gaussian noise. Note that the gradient of $\hat{U}$ is given by

$$\frac{\partial \hat{U}}{\partial \theta_i}(\boldsymbol{\theta}, \boldsymbol{\kappa}) = -\sum_{k=1}^{K} \left(S_{i,k} \frac{1}{\theta_i} - F_{i,k} \frac{\kappa_k}{1 - \theta_i \kappa_k}\right) \qquad , \ \forall i \in [L],$$

$$\frac{\partial \hat{U}}{\partial \kappa_k}(\boldsymbol{\theta}, \boldsymbol{\kappa}) = -\sum_{i=1}^{L} \left(S_{i,k} \frac{1}{\kappa_k} - F_{i,k} \frac{\theta_i}{1 - \theta_i \kappa_k}\right) \qquad , \ \forall k \in [K].$$

After these m consecutive steps, the algorithm returns a perturbed version of the last particle $(\boldsymbol{\theta}^{(m)}, \boldsymbol{\kappa}^{(m)})$.

Algorithm 9 requires the choice of three hyper-parameters: the number of steps m, h_0 which controls the size of the gradient steps, and γ which controls the uncertainty of the proposal $(\tilde{\boldsymbol{\theta}}, \tilde{\boldsymbol{\kappa}})$. These hyper-parameters are the same at each iteration of PB-LB. Regarding h_0 and γ, they are usually defined according to a smoothness property on the target function $\hat{U}$. In our setting, $\hat{U}$ is not smooth enough to derive theoretically founded values for h_0 and γ. However, we show in Section 4.2 that a careful tuning of both parameters drastically reduces the cumulative expected regret of PB-LB.

4.1.3 Overall complexity

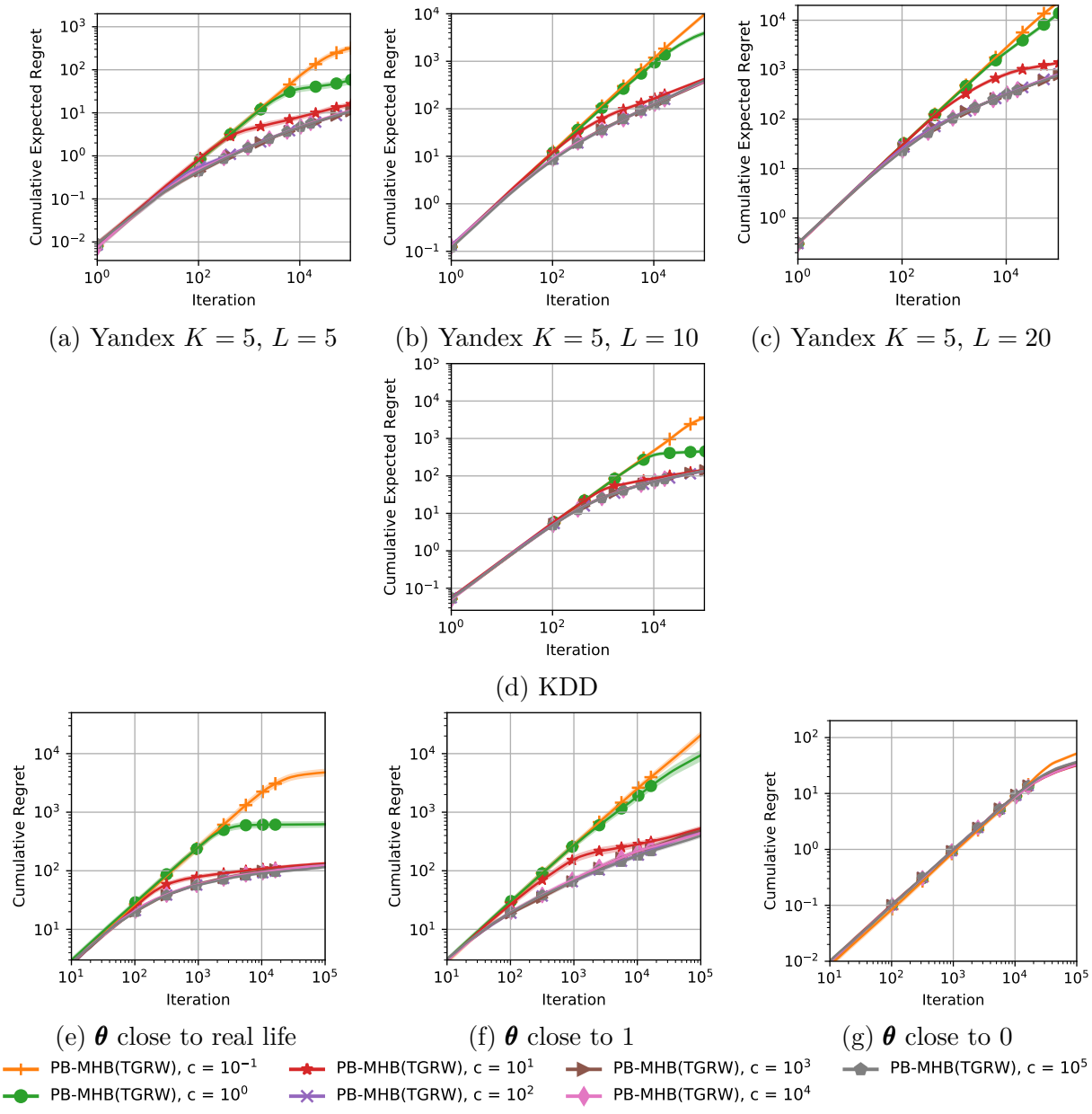
The computational complexity of PB-MHB (using Algorithm 8) is driven by the number of random-walk steps done per recommendation: $m(L + K - 1)$, which is controlled by the hyper-parameter m. We have a similar situation with PB-LB (using Algorithm 9), with m the number of steps of the gradient descent: the complexity per iteration is $\mathcal{O}(m(L + K))$. The hyper-parameter m corresponds to the burning period: the number of steps required by MCMC methods to draw a point $(\boldsymbol{\theta}^{(m)}, \boldsymbol{\kappa}^{(m)})$ almost independent from the initial one. While the requirement for a burning period may refrain us from using such methods in recommendation settings, we demonstrate in the following experiments that the required value for m remains reasonable. We drastically reduce m by starting the MCMC approximations call from the point used to recommend at previous iteration. This corresponds to the second option at Line 1 in Algorithm 8, and to the default behavior of Algorithm 9.

As we will see in Table 4.2, despite similar complexity, PB-MHB is much slower than PB-LB. This is partially due to the rejection phase of TGRW and to implementation details: the steps in Algorithm 9 are vectorized, while the steps in Algorithm 8 are sequential.

4.2 Practical results

In this section we demonstrate the benefit of the proposed approaches both on three artificial datasets called purely simulated and two real-life datasets (Yandex and KDD). All these datasets are presented in Section 1.3.4. In the Yandex setting, we look at the results averaged on the 10 most frequent queries, while displaying $K = 5$ items among the $L = 10$ most attractive ones. We also consider the settings $L = 5$ and $L = 20$. In the KDD setting, we look at the 8 most frequent queries, with L=3 and K depending on the query. Let us remind that whatever real-life data we are using, we use them to compute the parameters $\boldsymbol{\theta}$ and $\boldsymbol{\kappa}$ and simulate at each iteration a "real" user feedback (i.e. clicks) by applying PBM with these parameters. This is what is usually done in the literature since the recommendations done by a bandit are very unlikely to match the recommendations logged in the ground truth data and without matching, it would be impossible to compute a relevant reward for each interaction (see Section 1.3.3). We compare the performance of all versions of PB-MHB and PB-LB with the performance of TopRank [43] and GRAB [26] that were described respectively in Section 2.2.1 and Chapter 3.

We compare the algorithms on the basis of the *cumulative expected regret* (see Equation (4.8)), which is the sum, over T consecutive recommendations, of the difference between the expected reward of the best possible answer and of the answer of a given recommender



(a) Yandex $K = 5$, $L = 5$ (b) Yandex $K = 5$, $L = 10$ (c) Yandex $K = 5$, $L = 20$

(d) KDD

(e) $\boldsymbol{\theta}$ close to real life (f) $\boldsymbol{\theta}$ close to 1 (g) $\boldsymbol{\theta}$ close to 0

Figure 4.1: (TGRW) Cumulative expected regret w.r.t. the number of iterations on three datasets: Yandex ((a), (b) and (c)), KDD (d) and purely simulated ((e), (f) and (g)). Impact of the width $c/\sqrt{t}$ of Gaussian random-walk steps for the truncated proposal. The (narrow) shaded area depicts the standard error of our regret estimates.

system:

$$R_T \stackrel{def}{=} \sum_{t=1}^{T} \mathbb{E}\left[\sum_{k=1}^{K} r_k(t) \,\middle|\, \boldsymbol{a}(t) = \boldsymbol{a}^*\right] - \sum_{t=1}^{T} \mathbb{E}\left[\sum_{k=1}^{K} r_k(t) \,\middle|\, \boldsymbol{a}(t)\right] \tag{4.7}$$

$$= \mu^* T - \sum_{t=1}^{T} \sum_{k=1}^{K} \theta_{i_k(t)} \kappa_k. \tag{4.8}$$

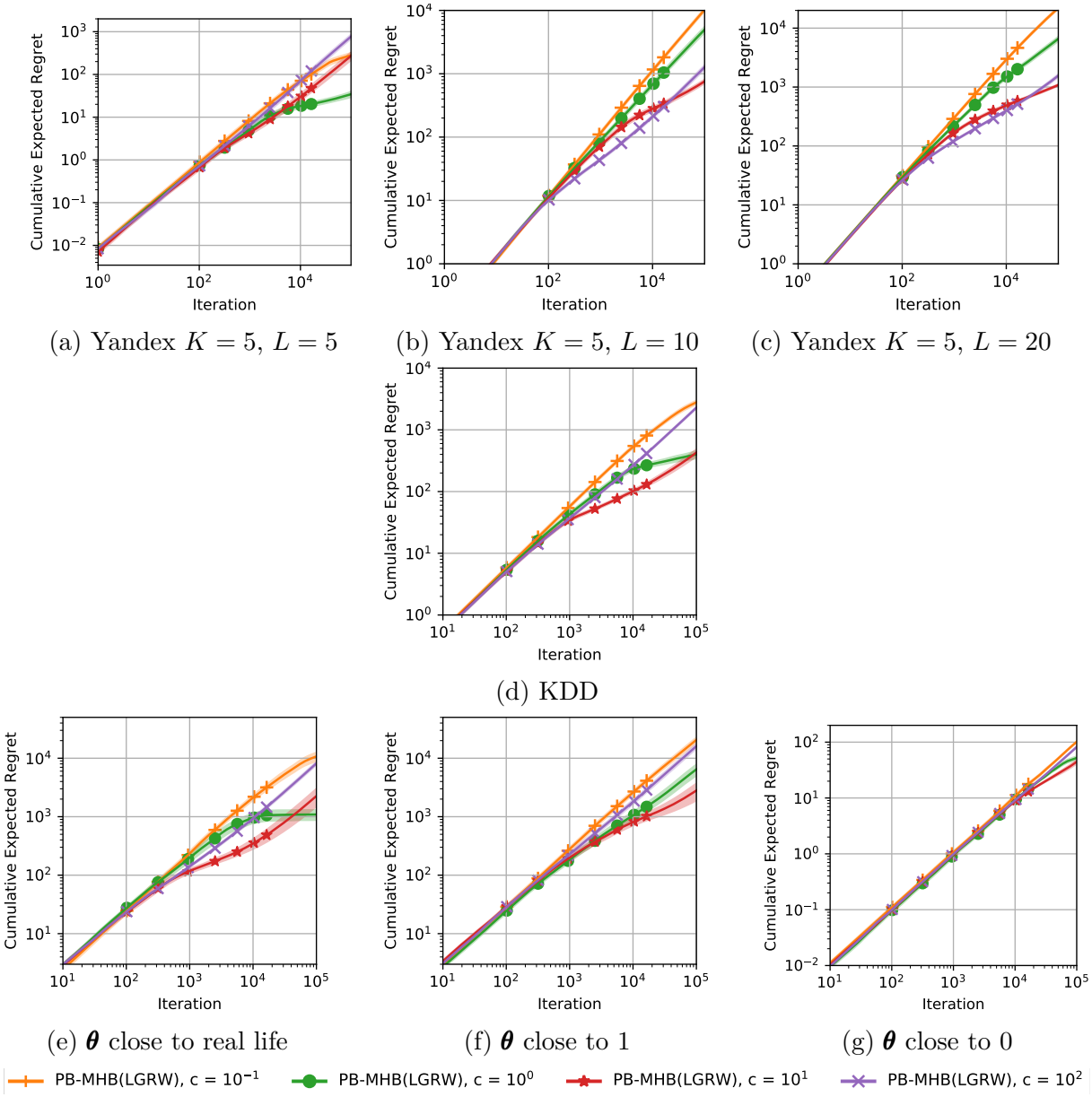
The regret is plotted with respect to T on a log-scale basis. The best algorithm is the one with the lowest regret. The log scale helps to identify the typical bandit behaviors. In particular, a linear tendency corresponds to an exploration phase, a constant tendency implies that the recommendation of the bandit matches the optimal recommendation, a log tendency means that the bandit recommendation is really close to the optimal recommendation and that the bandit is accurately learning, and finally an inflection is the sign that the bandit over exploit and that it takes wrong decisions. We average the results of each algorithm over 20 independent sequences of recommendations per query (in total: 20 sequences for simulated data, 160 sequences for KDD behavioral data and 200 sequences for Yandex behavioral data). The shaded area in the figures depicts the standard error of our regret estimates.

PB-MHB Hyper-Parameters

PB-MHB main hyper-parameter is the proposal law. We compare four proposals: Truncated Gaussian Random Walk (TGRW), Logistic Gaussian Random Walk (LGRW), MaxPos and PseudoView presented in section 4.1.1. The performance of PB-MHB is also impacted by the number m of Metropolis-Hastings steps per recommendation.

When PB-MHB uses a Gaussian Random Walk proposal (TGRW or LGRW), its behavior is affected by an additional hyper-parameter: the width $c/\sqrt{t}$ of the Gaussian random-walk steps. Figure 4.1 and Figure 4.2 show the impact of $c/\sqrt{t}$ on each dataset for both proposals. To measure the impact of $c/\sqrt{t}$ on each settings, we compare the cumulative expected regret of PB-MHB using TGRW (Figure 4.1) with $c \in [10^{-1}, 10^0, 10^1, 10^2, 10^3, 10^4, 10^5]$ and PB-MHB using LGRW (Figure 4.2) with $c \in [10^{-1}, 10^0, 10^1, 10^2]$.

Comparing both figures, we can conclude that the regret of the PB-MHB algorithm which uses the TGRW proposal is the smallest (compared to using LGRW) as soon as c is large enough ($c >= 100$), meaning that PB-MHB with TGRW proposal is learning fast and well. Even in extreme settings, where $\boldsymbol{\theta}$ is close to 0 or to 1, TGRW follows a



(a) Yandex $K = 5$, $L = 5$ (b) Yandex $K = 5$, $L = 10$ (c) Yandex $K = 5$, $L = 20$

(d) KDD

(e) $\boldsymbol{\theta}$ close to real life (f) $\boldsymbol{\theta}$ close to 1 (g) $\boldsymbol{\theta}$ close to 0

Figure 4.2: (LGRW) Cumulative expected regret w.r.t. the number of iterations on three datasets: Yandex ((a), (b) and (c)), KDD (d) and purely simulated ((e), (f) and (g)). Impact of the width $c/\sqrt{t}$ of Gaussian random-walk steps for the logit proposal. The shaded area depicts the standard error of our regret estimates.

log curve which means that PB-MHB is learning the best recommendation. We can see that when L is increasing while K is stable (Yandex $K = 5, L = 10$ and $K = 5, L = 20$), which means that when the set of available items is large compared to the number of position, the learning process is slower. The tunning of c for the LGRW proposal depends

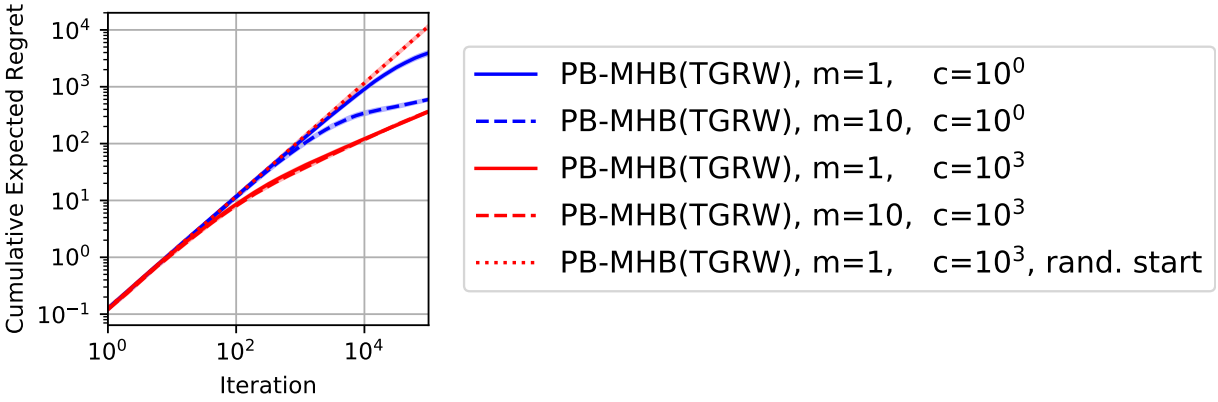
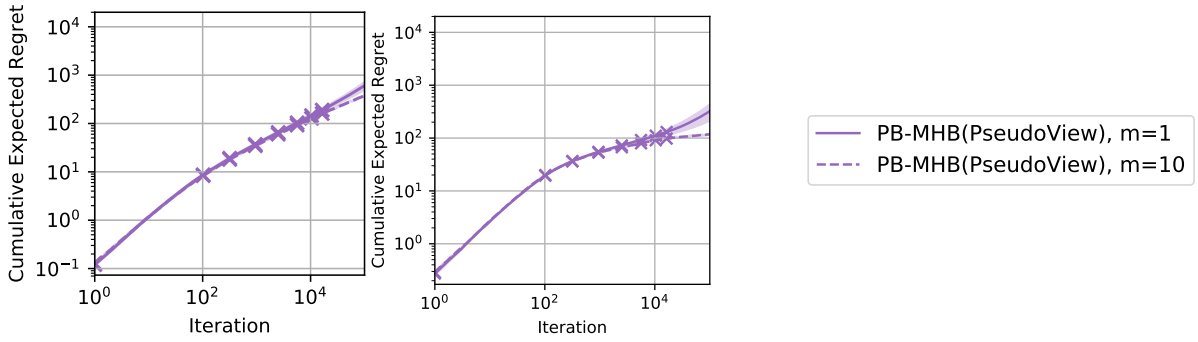


Figure 4.3: Cumulative expected regret w.r.t. iterations for TGRW proposal on Yandex dataset (with $K = 5$, $L = 10$). Impact of the use of the parameters from the previous iteration to warm-up the Metropolis-Hasting algorithm and of the number m of Metropolis-Hastings steps per recommendation. The results are computed with the TGRW proposal for $c = 1$ and $c = 10^3$. The shaded area depicts the standard error of our regret estimates.



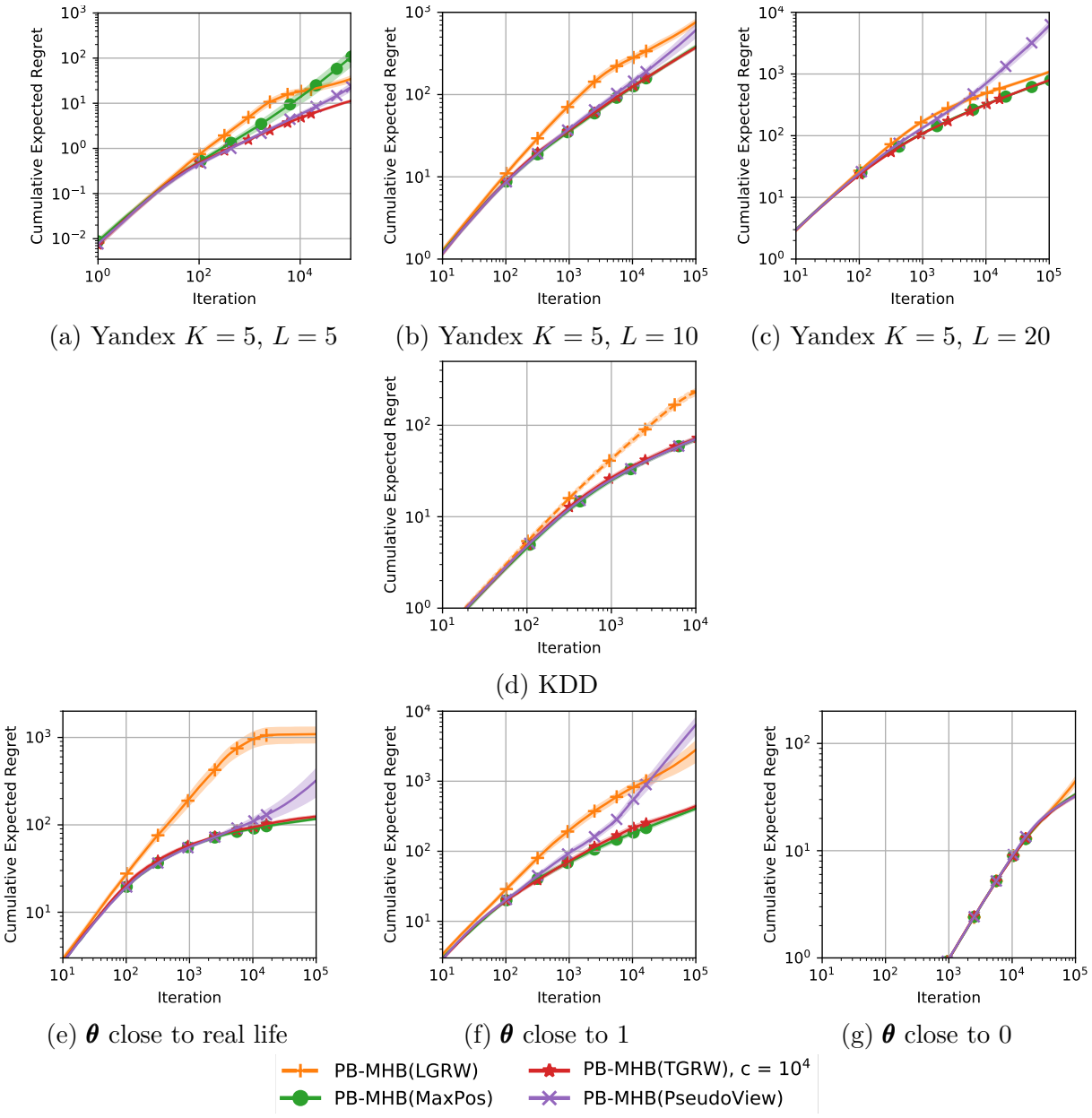
(a) Yandex $K = 5$, $L = 10$ (b) $\boldsymbol{\theta}$ close to real life

Figure 4.4: Cumulative expected regret w.r.t. iterations for PseudoView proposal on two datasets: (a) Yandex (with $K = 5$, $L = 10$) and (b) simulated data with $\boldsymbol{\theta}$ "close to real life". Impact of the number m of Metropolis-Hastings steps per recommendation. The shaded area depicts the standard error of our regret estimates.

more on the setting. If c is too small, PB-MHB explores too much and if c is higher than its optimal value, the curve of the cumulative expected regret is inflected meaning that the learning leads to a sub-optimal recommendation. Overall, TGRW gives more stable results (i.e. less sensitive to its hyperparameters) than LGRW, as it does not suffer from high value of c and reaches the best performances when $c >= 100$ for all settings.

In Figures 4.3 and 4.4, we illustrate the impact of m with TGRW and PseudoView proposals. Using both figures, we can see that increasing the number of steps improves the performance of PB-MHB. For TGRW (Figure 4.3), it yields a high regret only when

c and m are both low (blue solid curve): when the random-walk steps are too small the Metropolis-Hasting algorithm requires more steps to get uncorrelated samples $(\tilde{\boldsymbol{\theta}}, \tilde{\boldsymbol{\kappa}})$. For



(a) Yandex $K = 5$, $L = 5$ (b) Yandex $K = 5$, $L = 10$ (c) Yandex $K = 5$, $L = 20$

(d) KDD

(e) $\boldsymbol{\theta}$ close to real life (f) $\boldsymbol{\theta}$ close to 1 (g) $\boldsymbol{\theta}$ close to 0

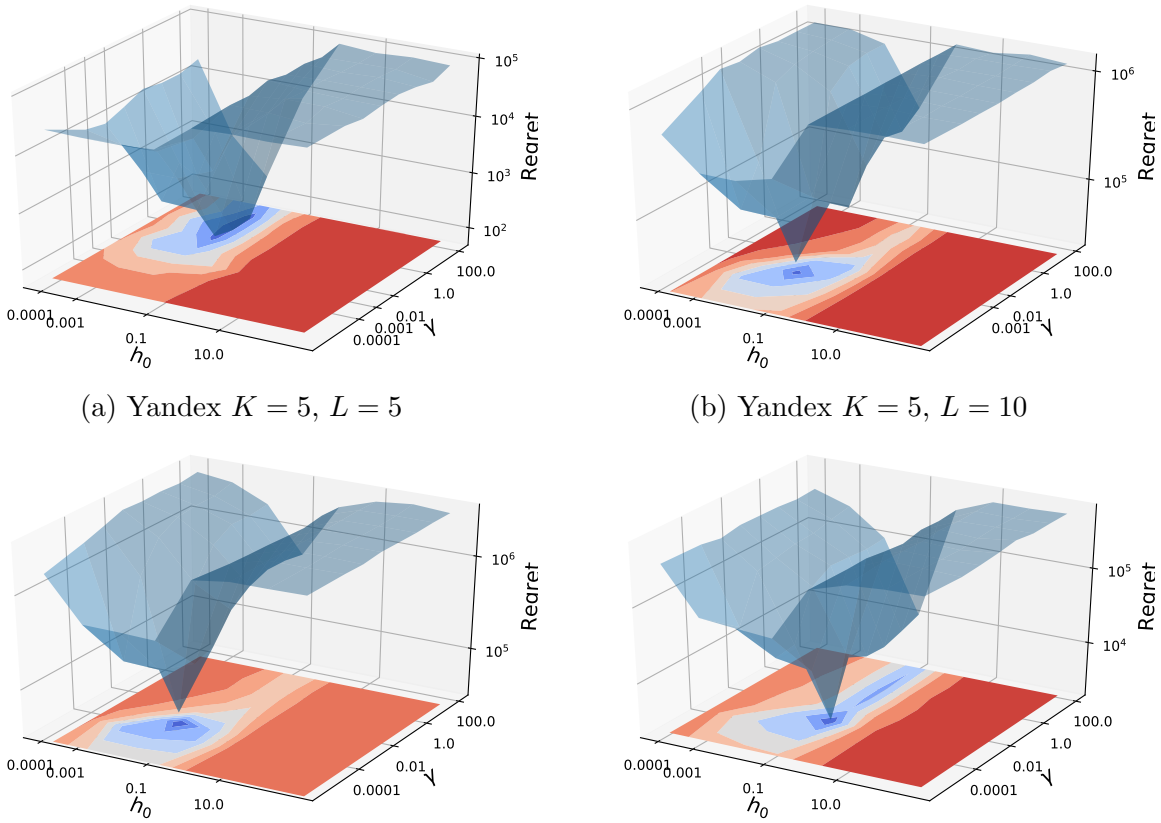
Figure 4.5: Cumulative expected regret w.r.t. the number of iterations on three datasets: Yandex ((a), (b) and (c)), KDD (d) and purely simulated ((e), (f) and (g)). Impact of the choice of the proposal for PB-MHB. c is set according to the best results of Figure 4.1, i.e. $c = 10^4$ for TGRW; For LGRW, $c = 10^0$ in (a), (d) and (e) and $c = 10^1$ in the other settings according to the best results of Figure 4.2.

reasonable values of c, m has no impact on the performance, which ease the tuning of PB-MHB hyper-parameters to obtain good recommendations. Overall, taking $c = 1000$ for the TGRW proposal and $m = 1$ is a good choice both in terms of regret and in terms of computation time, since the computation time of PB-MHB scales linearly with m. For PseudoView (Figure 4.4), we show the impact of m on datasets: Yandex with $K = 5, L = 10$ (Figure 4.4a) and the purely simulated settings with $\boldsymbol{\theta}$ close to real life (Figure 4.4b). We focus on these two settings in order to highlight the positive impact of increasing m on the learning process which can be seen in Figure 4.4a but easier to spot in Figure 4.4b. PseudoView has no additional parameter to correct the estimation of $\boldsymbol{\theta}, \boldsymbol{\kappa}$ for one step. Thus having $m = 10$ leads to better performances as the Markov chain can converge more accurately at each call. Thus, even when proposals with lower performance are chosen, PB-MHB can increase its performance by increasing m.

Furthermore the Metropolis-Hastings run of PB-MHB starts from the couple $(\tilde{\boldsymbol{\theta}}, \tilde{\boldsymbol{\kappa}})$ from the previous iteration. Figure 4.3 also shows the impact of keeping the parameters from the previous iteration compared to a purely random start. Note that this warm-up start allows PB-MHB to have a small regret while only doing $m = 1$ Metropolis-Hastings steps per recommendation. Starting from a new randomly drawn set of parameters would require more than $m = 10$ steps to obtain the same result, meaning a computation budget more than 10 times higher. This behavior is explained by the gap between the uniform law (which is used to draw the starting set of parameters) and the targeted law (*posterior* distribution of these parameters) which concentrates around its MAP. Even worse, this gap increases while getting more and more data since the *posterior* distribution concentrates with the increase of data. As a consequence, the required value for m increases along time when applying a standard Metropolis-Hasting initialisation, which explains why the dotted red line diverges from the solid one around iteration 10^4 in Figure 4.3.

Comparison of PB-MHB proposals

As shown in Figure 4.5, the TGRW proposal leads to the best results among all the proposal tested, while PseudoView and MaxPosition exhibit poor performances on some settings. LGRW has a higher regret than the other proposals and needs a proper tunning. MaxPosition has similar results as TGRW on most of the settings but performs very poorly on Yandex $K = 5, L = 5$. PseudoView has an inflected behavior on four settings out of seven which implies the need to increase m. As discuss in Section 4.1.3 this increase leads to unreasonable computation time due to the increase of the burning period.



(a) Yandex $K = 5$, $L = 5$

(b) Yandex $K = 5$, $L = 10$

(c) Yandex $K = 5$, $L = 20$

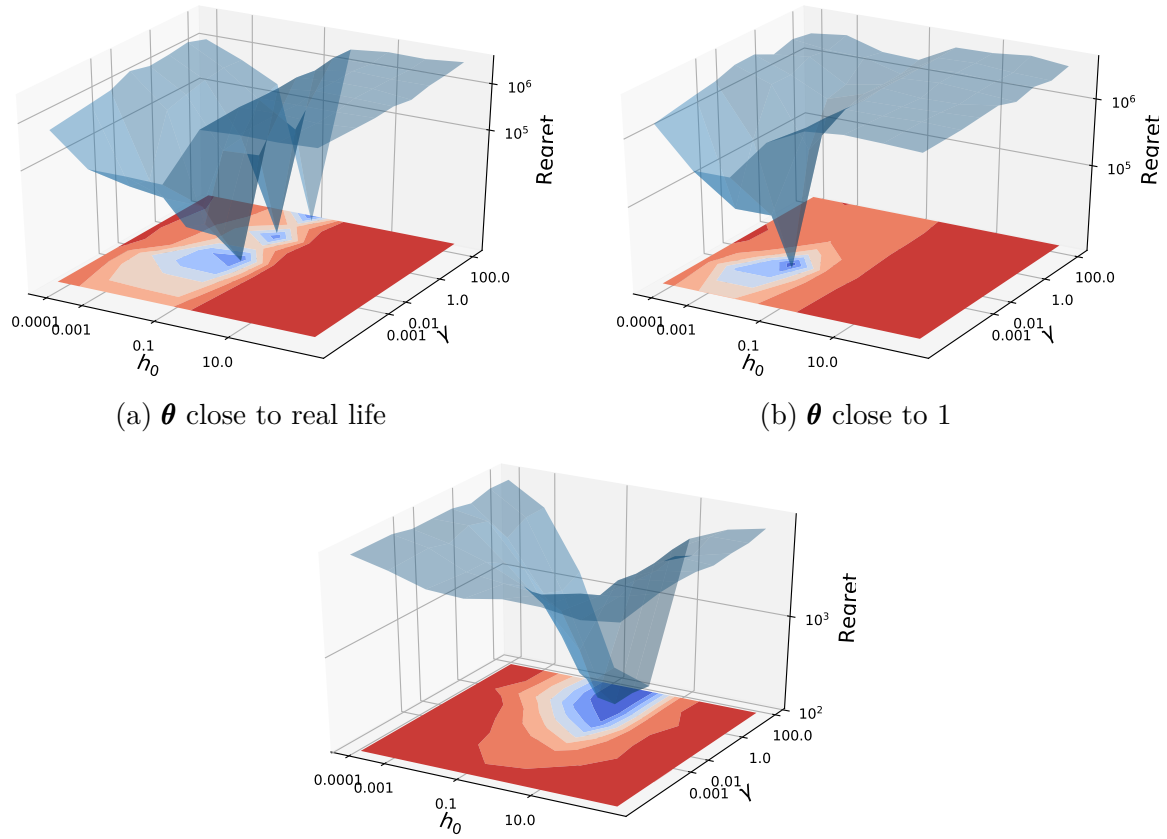
(d) KDD

Figure 4.6: Cumulative expected regret at T=10^7 for PB-LB on Yandex and KDD. The plotted surfaces correspond to the cumulative expected regret at iteration T=10^7 averaged over 20 independent sequences of recommendations per query (in total: 160 sequences for KDD and 200 sequences for each Yandex) for each couple of hyper-parameters (γ, h_0) used to tune PB-LB. Values of γ and h_0 scale axes y and x, the cumulative expected regret scales the axis z. Both 3D and 2D surfaces depict the impact of hyper-parameters γ and h_0 on the cumulative expected regret. The 2D surface is a heat map: blue regions show the lowest regret and red regions show the higher regret.

Thus, we selected PB-MHB with a TGRW proposal and $c = 10^4$ to compare with state-of-the-art algorithms.

PB-LB Hyper-Parameters

In Figure 4.6 and 4.7, we compare the performance of PB-LB depending on the value of its parameters γ and h_0. We take a large combination of values for γ and h_0 ranging respectively from 10^{-4} to 10^3 and from 10^{-6} to 10^2. The number of steps is set to $m = 1$



(a) $\boldsymbol{\theta}$ close to real life

(b) $\boldsymbol{\theta}$ close to 1

(c) $\boldsymbol{\theta}$ close to 0

Figure 4.7: Cumulative expected regret at T=10^7 for PB-LB on purely simulated settings. The plotted surfaces correspond to the cumulative expected regret at iteration T=10^7 averaged over 20 independent sequences of recommendations (in total: 20 sequences for each simulated data) for each couple of hyper-parameters (γ, h_0) used to tune PB-LB. Values of γ and h_0 scale axes y and x, the cumulative expected regret scales the axis z. Both 3D and 2D surfaces depict the impact of hyper-parameters γ and h_0 on the cumulative expected regret. The 2D surface is a heat map: blue regions show the lowest regret and red regions show the higher regret.

	Yandex			**KDD**	θ **close to**		
	$L = 5$	$L = 10$	$L = 20$		**real life**	**1**	**0**
h_0	0.001	0.001	0.001	0.01	0.01	0.1	0.001
γ	10.0	0.01	0.01	0.01	1000.0	100.0	00.1

Table 4.1: PB-LB hyper-parameters best values for $T = 10^7$ iterations on both behavioral and purely simulated settings.

as the experiments on PB-MHB showed (c.f. Fig. 4.3) the limited impact of m when other hyper-parameters are properly tuned, and as increasing m would increase the computational cost. In this experiment, we look at the cumulative expected regret at $T = 10^7$ iterations as a function of γ and h_0. We observe that both γ and h_0 require a fine tuning to get a low regret, otherwise the regret drastically increases. Moreover, these best combination of parameters depends on the setting (see. Table 4.1).

Comparison with Competitors

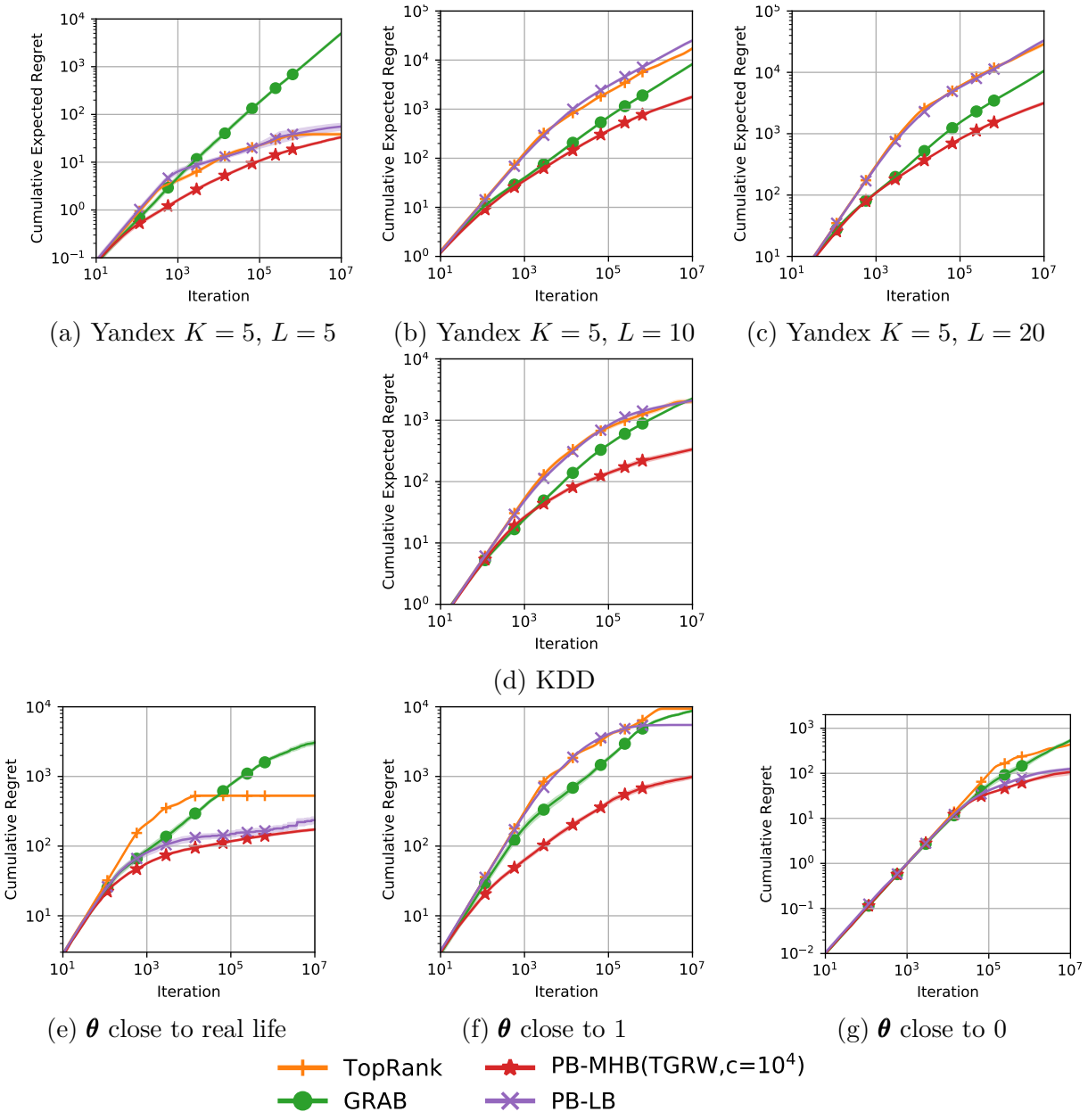
Figure 4.8 compares the regret obtained by PB-MHB, PB-LB, and their competitors on all our settings with various click and observation probabilities. In each setting, PB-MHB exhibits the smallest regret. On four settings out of seven, namely KDD, Yandex with $(K = 5, L = 10)$ and $(K = 5, L = 20)$ and purely simulated data with $\boldsymbol{\theta}$ close to 1, PB-MHB reduces the regret by an order of magnitude compared to its competitors, while on the other settings, Yandex with $(K = 5, L = 5)$ and purely simulated data with $\boldsymbol{\theta}$ close to 0 or close to real life, it is tightly followed by PB-LB or TopRank. The only behavioral setting on which the regret reduction is not by an order of magnitude is Yandex with $(K = 5, L = 5)$ which can be seen as a pure ordering task, this task being more adapted to TopRank. Even in this case, PB-MHB manages to have a lower regret than TopRank on most of the iterations. Overall, compared to its direct competitors, PB-MHB has the best regret performance on all the tested settings.

On the Yandex dataset various number, L, of available items are tested to show the impact of this number. As expected, increasing L increases the expected cumulative regret for PB-MHB, PB-LB and TopRank. GRAB is the least impacted by this variation with a regret at $T = 10^7$ between $5 * 10^3$ and 10^4 for all three Yandex settings. The increase of $L = 5$ to $L = 20$ on Yandex leads to an increase of 10^2 of PB-MHB's regret. For TopRank and PB-LB, this increase leads to a regret about 10^3 times higher.

Computation Time

Finally, in Table 4.2 we compare the computation time of each algorithm for different L on Yandex. PB-LB and GRAB are both stable as the increase of L does not impact their computation time per recommendation and they have the lowest computation time for higher L. TopRank is slower and its computation time increases with L. This may be due to the partition over all the items that TopRank has to compute to build its recommendation. Finally, despite its low regret, PB-MHB (with its best proposal, TGRW) is ten time slower

than the other algorithms in the setting $L = 5$ and its computation time increases with L.



(a) Yandex $K = 5$, $L = 5$ (b) Yandex $K = 5$, $L = 10$ (c) Yandex $K = 5$, $L = 20$

(d) KDD

(e) $\boldsymbol{\theta}$ close to real life (f) $\boldsymbol{\theta}$ close to 1 (g) $\boldsymbol{\theta}$ close to 0

Figure 4.8: Cumulative expected regret w.r.t. the number of iterations on three datasets: Yandex ((a), (b) and (c)), KDD (d) and purely simulated ((e), (f) and (g)) for all competitors. The plotted curves correspond to the average over 20 independent sequences of recommendations per query (in total: 20 sequences for simulated data, 160 sequences for KDD and 200 sequences for each Yandex). The shaded area depicts the standard error of our regret estimates. For PB-LB, h_0 and γ are set according to Table 4.1.

However, we believe that $30ms$ is still affordable for online experiences while the webpage is loading and this is still much lower (for better recommendations) than other algorithms (e.g. PMED [37] described in Section 2.1.1) proposed in the bandit-based RS literature.

Algorithm		**Computation Time (ms)**		
		$L = 5$	$L = 10$	$L = 20$
TopRank		0.7	3	7
GRAB		0.8	0.8	1
PB-LB		1	1	1
PB-MHB	TGRW, $c = 10^4$	11	17	30

Table 4.2: Average computation time per recommendation for a sequence of 10^7 recommendations vs. the first query of Yandex data, on an Intel Xeon E5640 CPU@2.67GHz with 50 GB RAM. The algorithms are implemented in Python.

4.3 Conclusion

We saw in this chapter that Metropolis Hasting gives the best practical result despite a slight loss in term of computational speed and its lack of theoretical proof. Langevin based method is also a good alternative but its tuning sensibility makes it quite hard to put in practice, when no information on users behavior are known. Now that we saw several method to handle PBM, we will see in the next chapter how the unimodal method can be extended to other click behavioral model.

Chapter 5

Unimodal bandits for other click behavioral models

Contents

5.1 Model assumption 93

5.2 UniRank: unimodal bandit algorithm for generic online ranking 95

5.3 Theoretical analysis 100

5.4 Practical results 103

5.5 Conclusion 105

In the two previous chapters, PBM is the main click behavioral model tackled. In this chapter, we reuse the Unimodal bandit framework seen in Chapter 3 to tackle other click behavioral model. This extension is more preferred to the extension of MCMC based bandits as it leads to theoretical guaranties through the upper confidence bound.

In this chapter, the setting tackled is an online learning to rank (OLR) problem with clicks feedback. The difference with previous chapters stands in the fact that the click model is not assume beforehand and thus ν can be refined to match any position-based assumptions (see Section 1.3 and Equation (1.1)*).*

The main contribution presented in this chapter is the new bandit algorithm, UniRank, dedicated to a generic online learning to rank setting. UniRank is inspired by unimodal bandit algorithms [13]: we implicitly consider a graph $\mathcal{G}$ on the partitions of the item-set $[L]$ such that the considered bandit setting is unimodal w.r.t. $\mathcal{G}$, and UniRank chooses each recommendation in the $\mathcal{G}$-neighborhood of an elicited partition. Thanks to this restricted exploration, UniRank is the first algorithm dedicated to a generic setting with a $O(L/\Delta \log T)$ regret upper-bound, while previous state-of-the-art algorithms were suffering a $O(LK/\Delta \log T)$ regret (see Table 5.1). Note that Unirank's upper-bound requires all items' attractiveness to be different, which is an usual assumption satisfied by real world

Algorithm	**Click model**	**Regret**	Δ $\theta_1 \geqslant \theta_2 \geqslant \cdots \geqslant \theta_L$
UniRank (our algorithm)	PBM*, CM*, ...	$\mathcal{O}\left(L/\Delta \log T\right)$	Detailed
	PBM, CM, ...	$\mathcal{O}\left(LK/\Delta \log T\right)$	in
UniRank (facing CM*)	CM*	$\mathcal{O}\left((L-K)/\Delta \log T\right)$	Chapter 5
TopRank [43]	PBM, CM, ...	$\mathcal{O}\left(LK/\Delta \log T\right)$	$\min_{(j,i)\in[L]\times[K]:j>i} \frac{\theta_i - \theta_j}{\theta_i}$
GRAB (Chapter 3)	PBM	$\mathcal{O}\left(\frac{L}{\Delta}\log T\right)$	$\min_{\boldsymbol{a}\in\mathcal{N}_{\pi^*}(\boldsymbol{a}^*)} \mu^* - \mu_{\boldsymbol{a}}$
PB-MHB (Chapter4)	PBM	unknown	$\emptyset$
PBM-PIE [41]	PBM	$\mathcal{O}\left((L-K)/\Delta \log T\right)$	$\min_{i\in\{K+1,\dots,L\}} \mu^* - \mu_{\boldsymbol{a}[K:=i]}$
CascadeKL-UCB [38]	CM	$\mathcal{O}\left((L-K)/\Delta \log T\right)$	$\min_{\boldsymbol{a}\in\mathcal{A}} \mu^* - \mu_{\boldsymbol{a}}$
SAM [58]	Matching*	$\mathcal{O}\left(L \log L/\Delta \log T\right)$	$\min_{\boldsymbol{a}\in\mathcal{N}_{\pi^*}(\boldsymbol{a}^*)} \mu^* - \mu_{\boldsymbol{a}}$
OSUB [13]	Unimodal	$\mathcal{O}\left(\gamma/\Delta \log T\right)$	$\min_{\boldsymbol{a}\in\mathcal{N}_G(\boldsymbol{a}^*)} \mu^* - \mu_{\boldsymbol{a}}$

Table 5.1: Required click model and upper-bound on cumulative regret for some well-known algorithms. The exact definition of Δ is specific to each algorithm and are defined in 3.1. The symbol * means Assumption 3, defined in Section 5.1, is satisfied.

applications. Table 5.1 also helps comparing regret upper bounds according to related algorithms' assumption on click model. The top three bound shows theoretical performances of UniRank under various click model assumption. UniRank achieves the lowest regret bound when facing multiple models. This bound reduces when restricting on CM and is the same as CascadeKL-UCB [38]. When facing PBM, PBM-PIE [41] has a lower regret bound but needs more information on the $\boldsymbol{\kappa}$ values as recall in Chapter 3.

From an application point of view, UniRank has several interesting features: it handles state of the art click models which have attraction-probabilities $\boldsymbol{\theta}$ altogether ; it is simple to implement and efficient in terms of computation time; it does not require the knowledge of the time horizon T; and it exhibits an empirical regret on par with other theoretically proven algorithms.

As an indirect contribution, UniRank demonstrates that unimodality is a key tool to analyze the intrinsic complexity of some combinatorial semi-bandit problems. We also demonstrate the flexibility of unimodal bandit algorithms and of the proof of their regret upper-bound. In particular, we extend [13]'s analysis to a graph which is unimodal in a weaker sens: (i) UniRank takes its decisions given an optimistic index which is not based on the expected reward but on the probability for an item to be more attractive than another one thanks to the comparison of clicks over items embodied by random variable $c_i(t)$ (see

Section 1.1) and (ii) some sub-optimal nodes in the handled graph have no better node in their neighborhood.

This chapter is organised as follow: Section 5.1 presents the assumption used to model our setting. We then introduce UniRank in Section 5.2, and theoretical guarantees and empirical performance are presented respectively in Section 5.3 and 5.4. We conclude in Section 5.5

5.1 Model assumption

Up to now, an OLR problem assumes two main properties: (i) a click at a position is a random variable only conditioned by the recommendation and the position, and (ii) the expectation of the corresponding distribution is fixed. We now introduce the three assumptions required by UniRank, which are fulfilled by PBM and CM click models.

We first assume an order on items.

Assumption 2 (strict weak order)**.** *There exits a* preferential attachment *function* $g : [L] \to \mathbb{R}$ *on items, and for any pair of items* (i, j)*,*

— *if* $g(i) > g(j)$*, item* i *is said* more attractive *than item* j*, which we denote* $i \succ j$*;*

— *if* $g(i) = g(j)$*, item* i *is said* equivalent *to item* j*, which we denote* $i \sim j$*.*

This assumption is an implicit assumption of state-of-the-art click-models (PBM, CM, *Dependent Click Model*). It ensures the existence of a strict weak order $\succ$ on items: the items may be ranked by attractiveness, some items being equivalent. A typical example with $L = 4$ would be $1 \succ 2 \sim 3 \succ 4$, meaning item 1 is more attractive than any other item, and items 2 and 3 are equivalent and more attractive than item 4. Such situation may also be represented with an ordered partition: $\{1\} \succ \{2, 3\} \succ \{4\}$, where $E \succ F$ means that for any item $i \in E$ and any item $j \in F$, $i \succ j$. In the rest of this chapter we will use either the preferential attachment function, or its associated strict weak order, or the corresponding ordered partition depending on the most appropriate representation.

The strongest results of the theoretical analysis require the slightly stronger assumption which ensures that two distinct items cannot be equivalent. This assumption is equivalent to any of both hypothesis: (i) the order $\succ$ is total, and (ii) each subset of the ordered partition is composed of only one item.

Assumption 3 (strict total order)**.** *There exits a* preferential attachment *function* $g : [L] \to \mathbb{R}$ *on items s.t. for any pair of distinct items* (i, j)*, either* $g(i) > g(j)$ *or* $g(j) > g(i)$*.*

Our next assumption indicates that recommending the items according to the order $\succ$ associated to the preferential attachment leads to an optimal recommendation.

Definition 2 (Compatibility with a strict weak order)**.** *Let $\succ$ be a strict weak order on the items, and $\boldsymbol{a}$ be a recommendation. The recommendation $\boldsymbol{a}$ is* compatible *with the order $\succ$ if*

1. *for any position $k \in [K-1]$, either $a_k \succ a_{k+1}$ or $a_k \sim a_{k+1}$;*
2. *for any item j in $[L] \setminus \boldsymbol{a}([K])$, either $a_K \succ j$ or $a_K \sim j$.*

Assumption 4 (Optimal reward)**.** *Any recommendation $\boldsymbol{a}$ compatible with $\succ$ is optimal, meaning $\mu_{\boldsymbol{a}} = \mu^*$.*

This assumption is of utmost importance for UniRank as it means that identifying a partition of the items which is coherent with $\succ$ is sufficient to ensure optimal recommendations. In the context of CM (respectively PBM) click model, this assumption means that the k-th most attractive item has to be placed at the k-th looked-at position (resp. the k-th most observed position), which obviously leads to the highest expected number of clicks.

Let us now consider the last assumption which regards the expectation of the random variable $c_i(t) - c_j(t)$, corresponded to the Expected click difference.

Definition 3 (Expected click difference)**.** *Let i and j be two items, and $\boldsymbol{a}$ a recommendation. The* probability of difference *and the* expected click difference *between items i and j w.r.t. the recommendation $\boldsymbol{a}$ are respectively:*

$$\tilde{\delta}_{i,j}(\boldsymbol{a}) = \mathbb{P}_{\boldsymbol{a}(t)\sim\mathcal{U}(\{\boldsymbol{a},(i,j)\circ\boldsymbol{a}\})}\left[c_i(t) \neq c_j(t)\right] \text{ and}$$
$$\tilde{\Delta}_{i,j}(\boldsymbol{a}) = \mathbb{E}_{\boldsymbol{a}(t)\sim\mathcal{U}(\{\boldsymbol{a},(i,j)\circ\boldsymbol{a}\})}\left[c_i(t) - c_j(t) \mid c_i(t) \neq c_j(t)\right],$$

where $(i,j)\circ\boldsymbol{a}$ is the permutation $\boldsymbol{a}$ such that items i and j have been swapped, and $\mathcal{U}(S)$ is the uniform distribution on the set S. If only i (respectively j) belongs to $\boldsymbol{a}$, $(i,j)\circ\boldsymbol{a}$ is the permutation $\boldsymbol{a}$ where item i is replaced by item j (resp. j by i). If neither i nor j belongs to $\boldsymbol{a}$, $(i,j)\circ\boldsymbol{a}$ is $\boldsymbol{a}$.

Assumption 5 (Order identifiability)**.** *The strict weak order $\succ$ on items is* identifiable*, meaning that for any couple of items (i,j) in $[L]^2$ s.t. $i \succ j$, and for any recommendation $\boldsymbol{a} \in \mathcal{P}_K^L$ s.t. at least one of both items is displayed, $\tilde{\delta}_{i,j}(\boldsymbol{a}) \neq 0$ and $\tilde{\Delta}_{i,j}(\boldsymbol{a}) > 0$.*

The expected click difference reflects the fact that an item leads to more clicks than another independently of the position of both items (other items being unchanged). Hence, Assumption 5 points out that when an item is more attractive than another one, it has a higher probability to be clicked upon, all other things being equal. This assumption is natural and ensures that the order on items may be recovered from the expected click difference, which can be observed.

Finally, the following lemma, proven in Appendix B.3, states that CM and PBM models fulfill our assumptions.

Lemma 4. *Let (L, K, ρ) be an online learning to rank problem with users following CM or PBM model with positions ranked by decreasing observation probabilities. Then Assumptions 2, 4, and 5 are fulfilled. Furthermore, Assumption 3 is fulfilled if for any couple of items, their attraction-probabilities θ differ.*

5.2 UniRank: unimodal bandit algorithm for generic online ranking

Our algorithm, UniRank, is detailed in Algorithm 10, and Figure 5.1 unfolds one iteration of UniRank. This algorithm takes inspiration from the unimodal bandit algorithm OSUB [13] by selecting at each iteration t an *arm to play* $\boldsymbol{P}(t)$ in the neighborhood of the current best one $\tilde{\boldsymbol{P}}(t)$ (a.k.a. the *leader*). However, UniRank's arms are not recommendations but sets of recommendations represented by ordered partitions. Hence, the recommendation $\boldsymbol{a}(t)$ is drawn uniformly at random in the subset $\mathcal{A}(\boldsymbol{P}(t))$ of recommendations compatible with $\boldsymbol{P}(t)$.

Let us now first define the notations used by UniRank and then present its concrete behaviour.

Statistic $\hat{s}_{i,j}(t)$ UniRank's choices are based on the statistic $\hat{s}_{i,j}(t)$ and pessimistic estimators of its expected value: the Kullback-Leibler-based one denoted $\underline{s}_{i,j}(t)$, and the slightly pessimistic one $\tilde{s}_{i,j}(t)$. $\hat{s}_{i,j}(t)$ is the average value of $c_i(s) - c_j(s)$ for s in $[t-1]$, where we restrict ourselves to iterations at which items i and j are in the same subset of the played partition $\boldsymbol{P}(s)$, and $c_i(s) \neq c_j(s)$. More specifically,

$$\hat{s}_{i,j}(t) \stackrel{def}{=} \frac{1}{T_{i,j}(t)} \sum_{s=1}^{t-1} O_{i,j}(s)(c_i(s) - c_j(s)),$$

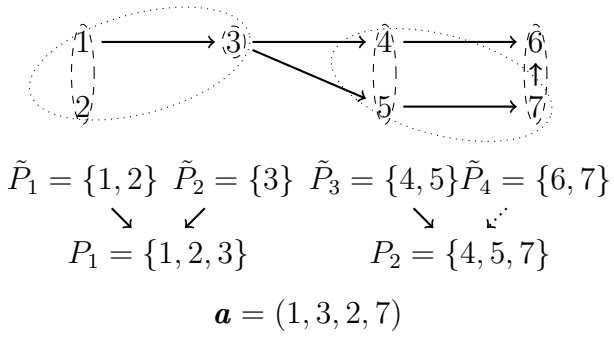


Figure 5.1: One iteration of UniRank with $L = 7$ items and $K = 4$ positions (t is omitted for clarity). Each arrow $i \to j$ in the top graph on items means the sligthly pessimistic statistic is non-negative ($\tilde{s}_{i,j} > 0$). With these values, the leader partition (represented with dashed ellipses) is $(\tilde{P}_1, \tilde{P}_2, \tilde{P}_3, \tilde{P}_4)$, where $\tilde{P}_4$ gathers remaining items as the 3 first partitions contain more than K items. Then, we assume that $\underline{s}_{1,3} \leqslant 0$, $\underline{s}_{3,4} \leqslant 0$, and $\underline{s}_{5,7} < \underline{s}_{5,6} \leqslant 0$ and we represent with dotted ellipses the corresponding played partition (P_1, P_2). P_1 derives form the merge of $\tilde{P}_1$ and $\tilde{P}_2$ as item 3 is not clearly less attractive than item 1. $\tilde{P}_2$ and $\tilde{P}_3$ are not merged as $\tilde{P}_2$ is already merged with its predecessor. Last, P_2 is obtained by adding item 7 from $\tilde{P}_4$ to $\tilde{P}_3$ as item 5 is not clearly less attractive than item 7, and item 7 is the best item having this property. Finally the recommendation $\boldsymbol{a}$ is obtained by concatenating a random permutation of P_1 with a random permutation of 1 item from P_2.

where $O_{i,j}(s) \overset{def}{=} \mathbb{1}\left\{\exists c, (i,j) \in P_c(s)^2\right\} \mathbb{1}\{c_i(s) \neq c_j(s)\}$ denotes that the difference between items i and$j$ is observable at iteration s, $T_{i,j}(t) \overset{def}{=} \sum_{s=1}^{t-1} O_{i,j}(s)$, and $\hat{s}_{i,j}(t) \overset{def}{=} 0$ when $T_{i,j}(t) = 0$.

The statistics $\hat{s}_{i,j}(t)$ are paired with their respective pessimistic *indices*

$$\underline{s}_{i,j}(t) \overset{def}{=} 2 * f\left(\frac{1+\hat{s}_{i,j}(t)}{2}, T_{i,j}(t), \tilde{t}_{\tilde{\boldsymbol{P}}(t)}(t)\right) - 1,$$

where f is a function from $[0,1] \times \mathbb{N} \times \mathbb{N}$ to $[0,1]$ and $f(\hat{\mu}, T, t) \overset{def}{=} \inf\{\mu \in [0, \hat{\mu}] : T \times \mathrm{kl}(\hat{\mu}, \mu) \leq \log(t) + 3\log(\log(t))\}$, with $\mathrm{kl}(p,q) \overset{def}{=} p\log\left({}^{p}/_{q}\right) + (1-p)\log\left({}^{1-p}/_{1-q}\right)$ the *Kullback-Leibler divergence* (KL) from a Bernoulli distribution of mean p to a Bernoulli distribution of mean q; $f(\hat{\mu}, T, t) \overset{def}{=} 0$ when $\hat{\mu} = 1$, $T = 0$, or $t = 0$; and $\tilde{t}_{\tilde{\boldsymbol{P}}(t)}$ is the number of iterations the leader $\tilde{\boldsymbol{P}}(t)$ at iteration t has previously been the leader. This pessimistic index is similar to the one used for KL-based bandit algorithms, after a rescaling of $\hat{s}_{i,j}(t)$ to the interval $[0,1]$.

Finally, UniRank also make use of the slightly pessimistic estimate $\tilde{s}_{i,j}(t) \overset{def}{=} \hat{s}_{i,j}(t) - \sqrt{\log\log t / T_{i,j}(t)}$.

Algorithm 10 UniRank: Unimodal Bandit Algorithm for Generic Online Ranking

Require: number of items L, number of positions K

for $t = 1, 2, \ldots$ **do**

  {leader-partition elicitation}

  construct $\tilde{P}(t)$ using Leader-partition elicitation method (see Algorithm 11)

  {optimistic partition elicitation}

  $\tilde{c} \leftarrow 1$; $d \leftarrow 0$

  while $\tilde{c} \leq \tilde{d} - 2$ **do**

    $d \leftarrow d + 1$

    if $\min_{(i,j) \in \tilde{P}_{\tilde{c}}(t) \times \tilde{P}_{\tilde{c}+1}(t)} \underline{s}_{i,j}(t) < 0$ **then**

      {merge both subsets}

      $P_d(t) \leftarrow \tilde{P}_{\tilde{c}}(t) \cup \tilde{P}_{\tilde{c}+1}(t)$; $\tilde{c} \leftarrow \tilde{c} + 2$

    else

      {keep current subset untouched}

      $P_d(t) \leftarrow \tilde{P}_{\tilde{c}}(t)$; $\tilde{c} \leftarrow \tilde{c} + 1$

    end if

  end while

  if $\tilde{c} = \tilde{d} - 1$ **then**

    $d \leftarrow d + 1$; $P_d(t) \leftarrow \tilde{P}_{\tilde{d}-1}(t)$

    if $\tilde{P}_{\tilde{d}}(t) \neq \varnothing$ **and** $\min\limits_{(i,j) \in \tilde{P}_{\tilde{d}-1}(t) \times \tilde{P}_{\tilde{d}}(t)} \underline{s}_{i,j}(t) < 0$ **then**

      {add best item from remaining ones}

      $P_d(t) \leftarrow P_d(t) \cup \left\{ \operatorname{argmin}\limits_{j \in \tilde{P}_{\tilde{d}}(t)} \min\limits_{i \in \tilde{P}_{\tilde{d}-1}(t)} \underline{s}_{i,j}(t) \right\}$

    end if

  end if

  $d(t) \leftarrow d$

  {recommendation}

  choose $\boldsymbol{a}(t)$ uniformly at random in $\mathcal{A}(\boldsymbol{P}(t))$

  observe the clicks vector $\boldsymbol{c}(t)$

end for

Algorithm 11 Leader-partition elicitation

Require: number of items L, number of positions K

$\tilde{d} \leftarrow 0$; $R \leftarrow [L]$

repeat

$\quad B \leftarrow \left\{ j \in R : \forall i \in R, \hat{s}_{i,j}(t) - \sqrt{\frac{\log\log t}{T_{i,j}(t)}} < 0 \right\}$

$\quad$ **if** $B \neq \varnothing$ **then** $\tilde{d} \leftarrow \tilde{d} + 1$; $\tilde{P}_{\tilde{d}}(t) \leftarrow B$ **end if**

until $R = \varnothing$ **or** $B = \varnothing$ **or** $\left|\bigcup_{\tilde{c}=1}^{\tilde{d}} \tilde{P}_{\tilde{c}}(t)\right| \geqslant K$

if $R \neq \varnothing$ **then** $\tilde{d} \leftarrow \tilde{d} + 1$; $\tilde{P}_{\tilde{d}}(t) \leftarrow R$ **end if**

return $\tilde{P}(t)$

Leader Elicitation At each iteration, UniRank first builds a partition $\tilde{\boldsymbol{P}}(t) = (\tilde{P}_1(t), \ldots, \tilde{P}_{\tilde{d}}(t))$ which is *coherent* with $\tilde{s}_{i,j}(t)$, meaning that for any couple of items (i, j) in $[L]^2$, if $\tilde{s}_{i,j}(t) > 0$ then either i belongs to a subset $\tilde{P}_c(t)$ ranked before the subset of j, or there exists a cycle $(i_1, i_2, \ldots, i_N)$ such that $i_1 = i_N = i$, $i_2 = j$, and for any $n \in [N-1]$, $\tilde{s}_{i_n, i_{n+1}}(t) > 0$. This partition is iteratively build by repeating the process of (i) gathering in a subset the non-dominated items (meaning the items j for which $\tilde{s}_{i,j}(t) < 0$ for any remaining item i), and (ii) removing them. A special care is taken to handle situations with cycles, and to gather in the same subset remaining items as soon as the first subsets contain more than K items.

Optimistic Partition Elicitation The partition $\tilde{\boldsymbol{P}}(t)$ plays the role of leader, meaning that at each iteration, UniRank picks a permutation $\boldsymbol{P}(t)$ in the neighborhood $\mathcal{N}(\tilde{\boldsymbol{P}}(t))$ of $\tilde{\boldsymbol{P}}(t)$, solving an exploration-exploitation dilemma. The partition $\boldsymbol{P}(t)$ is build by merging consecutive subsets $\tilde{P}_c(t)$ and $\tilde{P}_{c+1}(t)$ of the partition $\tilde{\boldsymbol{P}}(t)$, where $c \in [d-2]$, if one of the items in $\tilde{P}_{c+1}(t)$ is not clearly less attractive than all items in $\tilde{P}_c(t)$. The difference in attractiveness is measured after the pessimistic estimator $\underline{s}_{i,j}(t)$. Note that the subset $\tilde{P}_{\tilde{d}}(t)$ is never merged with $\tilde{P}_{\tilde{d}-1}(t)$, only the arm j in $\tilde{P}_{\tilde{d}}(t)$ with the smallest estimate $\min_{i \in \tilde{P}_{\tilde{d}-1}(t)} \underline{s}_{i,j}(t)$ is added to $\tilde{P}_{\tilde{d}-1}(t)$ if this estimate is non-positive.

Remark 6 (Recommendation chosen at random)**.** *Taking a random permutation is required to control the statistic $\hat{s}_{i,j}(t)$. Indeed, the analysis requires the probability for i to be ranked before j in the recommendation to be even. Overall, the aim is to identify a partition $\boldsymbol{P}^*$ such that any permutation in $\mathcal{A}(\boldsymbol{P}^*)$ is compatible with the unknown strict weak order on items.*

Remark 7 (Leader chosen given a pessimistic estimator)**.** *Note that UniRank uses a slightly pessimistic estimator to choose the leader, while OSUB selects the leader based on a maximum likelihood estimator. From a theoretical point of view, both criteria are equivalent, but the increase in stability brought by the pessimistic estimator drastically reduces the regret suffered by UniRank in practice.*

Remark 8 (Last subset of $\tilde{\boldsymbol{P}}(t)$)**.** *To keep the algorithm and its analysis simple, the partition $\tilde{\boldsymbol{P}}(t)$ is such that $\sum_{c=1}^{\tilde{d}-2} |\tilde{P}_c| < K \leqslant \sum_{c=1}^{\tilde{d}-1} |\tilde{P}_c|$, meaning that the items in $\tilde{P}_{\tilde{d}}(t)$ are the one which are never displayed by the recommendations in $\mathcal{A}\left(\tilde{\boldsymbol{P}}(t)\right)$. The subset $\tilde{\boldsymbol{P}}_{\tilde{d}}(t)$ may be empty.*

5.3 Theoretical analysis

The proof of the upper-bound on the regret of UniRank follows a similar path as the proof of OSUB [13]: (i) apply a standard bandit analysis to control the regret under the condition that the leader $\tilde{\boldsymbol{P}}(t)$ is an optimal partition, and (ii) upper-bound by $\mathcal{O}(\log\log T)$ the expected number of iterations such that $\tilde{\boldsymbol{P}}(t)$ is not an optimal partition.

However, both steps differ from [13]. First, UniRank handles partitions instead of recommendations. Secondly, it builds upon $\hat{s}_{i,j}(t)$ instead of estimators of the expected reward. While $\hat{s}_{i,j}(t)$ is the average of dependent random variables with different expected values, these expected values are greater than some non-negative constant $\tilde{\Delta}_{i,j}$ when $i \succ j$, which is sufficient to lower-bound $\hat{s}_{i,j}(t)$ away from 0 as required by the proof of the regret upper-bound (see Appendices B.4.2, B.4.3, and B.4.4 for details). Thirdly, the proof is adapted to handle the fact that $T_{i,j}(t)$ randomly increases when we play items i and j due to the exploration-exploitation rule, which is unusual in the bandit literature. Finally, while at each iteration UniRank is allowed to apply several simultaneous merges of two partitions, we prove that the regret is the same as when at most one merge is done per iteration. Up to our knowledge, this exploration-exploitation strategy and its analysis are new in the bandit community. We believe that it opens new perspectives for other semi-bandit settings.

Note that, as in [58], we restrict the theoretical analysis to the setting where the order on items is total, meaning we use Assumption 3. Without loss of generality, we also assume that $1 \succ 2 \succ \cdots \succ L$. Hence the only partition $\boldsymbol{P}^*$ which is such that, any permutation $\boldsymbol{a}$ in $\mathcal{A}(\boldsymbol{P}^*)$ is compatible with the unknown strict total order on items, is $(\{1\}, \dots \{K\}, [L] \setminus [K])$.

We now propose the main theorem that upper-bounds the regret of UniRank.

Theorem 3 (Upper-bound on the regret of UniRank assuming a total order on items)**.** *Let (L, K, ρ) be an OLR problem satisfying Assumptions 3, 4, and 5 and such that $1 \succ 2 \succ \cdots \succ L$. Denoting $\boldsymbol{P}^* = (\{1\}, \ldots, \{K\}, [L] \setminus [K])$ the optimal partition associated to this order, when facing this problem, UniRank fulfills*

$$\forall k \in [L] \setminus \{1\},\ \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\left\{ \begin{smallmatrix} \tilde{\boldsymbol{P}}(t)=\boldsymbol{P}^*, \\ \exists c, P_c(t)=\{\min(k-1,K),k\} \end{smallmatrix} \right\}\right] \leqslant \frac{16}{\tilde{\delta}_k^* \tilde{\Delta}_k^2} \log T + \mathcal{O}\left(\log\log T\right), \quad (5.1)$$

$$\text{and } \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{\tilde{\boldsymbol{P}}(t) \neq \boldsymbol{P}^*\}\right] = \mathcal{O}\left(\log\log T\right), \quad (5.2)$$

and hence

$$R(T) \leqslant \sum_{k=2}^{L} \frac{8\Delta_k}{\tilde{\delta}_k^* \tilde{\Delta}_k^2} \log T + \mathcal{O}\left(\log\log T\right) = \mathcal{O}\left(\frac{L}{\Delta} \log T\right),$$

where for any position $k > 1$, denoting $\ell \overset{def}{=} \min(k-1, K)$,
$\tilde{\delta}_k^* \overset{def}{=} \min_{\boldsymbol{P} \in \mathcal{N}(\boldsymbol{P}^*): \exists c, (\ell,k) \in P_c^2} \mathbb{P}_{\boldsymbol{a}(t) \sim \mathcal{U}(\mathcal{A}(\boldsymbol{P}))}\left[c_\ell(t) \neq c_k(t)\right]$,
$\tilde{\Delta}_k \overset{def}{=} \min_{\boldsymbol{a} \in \mathcal{P}_K^L: \{\ell,k\} \cap \boldsymbol{a}([K]) \neq \varnothing} \tilde{\Delta}_{\ell,k}(\boldsymbol{a})$,
$\Delta_k \overset{def}{=} \mu_{(1,\ldots,K)} - \mu_{(\ell,k)\circ(1,\ldots,K)}$,
$\Delta \overset{def}{=} \min_{k \in \{2,\ldots,L\}} \tilde{\delta}_k^* \tilde{\Delta}_k^2 / \Delta_k$, and $\mathcal{N}(\boldsymbol{P}^*)$ the set of partitions in the neighbor of $\boldsymbol{P}^*$.

The first upper-bound (Equation (5.1)) controls the expected number of iterations at which UniRank explores while the leader is the optimal partition. Both types of exploration are coverred: the merging of two consecutive subsets of $\tilde{\boldsymbol{P}}(t)$, and the addition of a sub-optimal arm to the last subset of the chosen partition $\boldsymbol{P}(t)$. The second upper-bound (Equation (5.2)) regards the expected number of iterations at which the leader is not the optimal partition. Let us now express the same bounds assuming the state of the art click models.

Corollary 1 (Facing CM* click model)**.** *Under the hypotheses of Theorem 3, if the user*

follows CM with probability θ_i to click on item i when it is observed, then UniRank fulfills

$$\begin{aligned} R(T) &\leqslant \sum_{k=K+1}^{L} 16\frac{\theta_K+\theta_k}{\theta_K-\theta_k}\log T + \mathcal{O}\left(\log\log T\right) \\ &= \mathcal{O}\left((L-K)\frac{\theta_K+\theta_{K+1}}{\theta_K-\theta_{K+1}}\log T\right). \end{aligned}$$

Corollary 2 (Facing PBM* click model)**.** *Under the hypotheses of Theorem 3, if the user follows PBM with the probability θ_i of clicking on item i when it is observed and the probability κ_k of observing the position k, then UniRank fulfills*

$$R(T) = \mathcal{O}\left(\frac{L}{\Delta}\log T\right),$$

where $\Delta \overset{def}{=} \min\{\min_{k\in\{K+1,\dots,L\}} \frac{1}{2}\frac{\theta_K-\theta_k}{\theta_K+\theta_k}, \min_{k\in\{2,\dots,K\}} \frac{(\frac{1}{2}(\kappa_{k-1}+\kappa_k)(\theta_{k-1}+\theta_k)-2\kappa_{k-1}\kappa_k\theta_{k-1}\theta_k)(\theta_{k-1}-\theta_k)}{(\kappa_{k-1}-\kappa_k)(\theta_{k-1}+\theta_k)^2}\}$.

Note that the regret upper-bound reduces to $\mathcal{O}\left((L-K)/\Delta\log T\right)$ with CM since, with this click model, the recommendation is optimal as soon as no sub-optimal item is displayed.

A more detailed version of these corollaries is given in the appendix, together with their proofs and Theorem 3's proof. These proofs builds upon the following pseudo-unimodality property which we also prove in the appendix.

Lemma 5 (Pseudo-unimodality assuming a total order on items)**.** *Under the hypotheses of Theorem 3, for any ordered partition of the items $\tilde{\boldsymbol{P}} = \left(\tilde{P}_1,\dots,\tilde{P}_{\tilde{d}}\right) \neq \boldsymbol{P}^*$,*

- *either there exists $c \in [\tilde{d}-1]$ such that $|\tilde{P}_c| > 1$;*
- *or $\tilde{P}_1 = \{i\}$ and there exists $j \in \tilde{P}_2$ such that $j \succ i$;*
- *or there exists $c \in [\tilde{d}-1]\setminus\{1\}$ such that $\tilde{P}_{c-1} = \{i'\}$, $\tilde{P}_c = \{i\}$, $i' \succ i$, and there exists $j \in \tilde{P}_{c+1}$ such that $j \succ i$.*

The first alternative implies that the subset $\tilde{P}_c$ should be split, which will be discovered by recommending permutations compatible with either $\tilde{\boldsymbol{P}}$ or its neighbors. Both other alternatives imply that j should be in a subset ranked before the subset containing i, which will discovered by recommending permutations compatible with any neighbor of $\tilde{\boldsymbol{P}}$ obtained by merging $\{i\}$ with the subsequent subset.

Remark 9 (Upper-bound on the regret assuming a strict total order on items)**.** *In [58],*

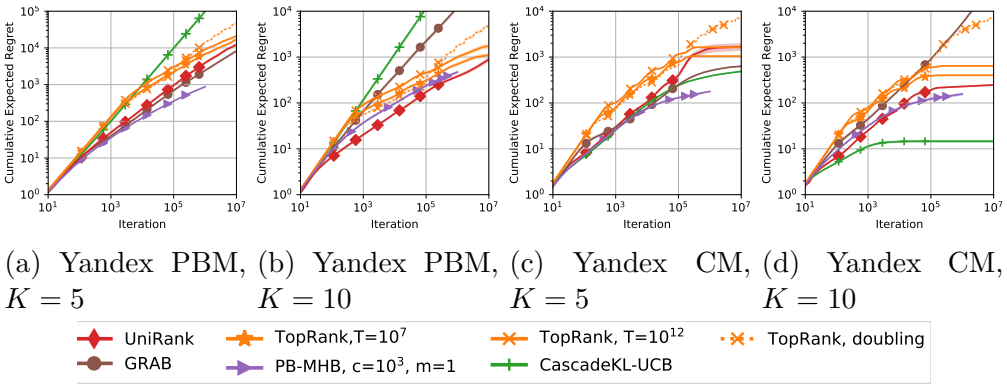
to prove a $\mathcal{O}(L \log L/\Delta \log T)$ regret-bound for a setting related to learning to rank, items are also assumed to have strictly different attractiveness.

Remark 10 (Upper-bound on the regret of UniRank assuming a weak order on items)**.** *If the order on items is not total, the proof of Theorem 3 may be adapted to get a $\mathcal{O}(LK/\Delta \log T)$ bound. Indeed, under the strict weak order assumption, there exists a set of optimal partitions, and therefore, any permutation compatible with a neighbor of any of these partitions may be recommended $\mathcal{O}(1/\Delta \log T)$ times. In the worst case scenario, K items are equivalent and strictly more attractive than the $L-K$ remaining items, and the set of the permutations compatible with a neighbor partition is composed of $K(L-K)$ permutations, which translates into a $\mathcal{O}(LK/\Delta \log T)$ regret bound. Note that [43] proves a $\Omega(LK/\Delta \log T)$ lower-bound on the regret assuming that the best items have the same attractiveness which means that the upper-bound of UniRank for this specific setting is optimal.*

5.4 Practical results

In this section, we compare UniRank to TopRank [43], PB-MHB (Chapter 4), GRAB (Chapter 3), and CascadeKL-UCB [38]. The experiments are conducted on the Yandex dataset, detailed in Section 1.3.4 both in PBM and CM settings. We look at the results averaged on the 10 most frequent queries, while displaying $K = 10$ or $K = 5$ items among the $L = 10$ most attractive ones selected among all items possible for each query, we then observe the regret over the top 5 positions. Having various K values allow us to test the impact of the quantity of information given to each algorithm. We use the cumulative regret (see Equation (1.6)) to evaluate the performance of each algorithm, where the cumulative regret is averaged over 20 independent runs per selected query of T= 10^7 iterations each (see the source code in the supplementary material). CascadeKL-UCB [38] is an algorithm dedicated to CM setting in which it shows a regret in $\mathcal{O}((L-K)/\Delta \log T)$.

Our results are shown in Figure 5.2. As expected, CascadeKL-UCB outperforms other algorithms in the CM model for which it is designed and suffers a linear regret (i.e. very high) in the PBM model. Surprisingly, although PB-MHB and GRAB are designed for PBM model, (i) PB-MHB is the first or second best algorithm in all models and even outperforms CascadeKL-UCB for $K = 5$ in CM model, and (ii) GRAB is ranked second and third in all models when $K = 5$. We conjecture that the good results of PB-MHB and GRAB in the CM model results from CM model being equivalent to a PBM model in



(a) Yandex PBM, $K = 5$ (b) Yandex PBM, $K = 10$ (c) Yandex CM, $K = 5$ (d) Yandex CM, $K = 10$

Figure 5.2: Cumulative regret on the 5 first positions w.r.t. iterations on Yandex dataset with $L = 10$ and $K = 5$ or $K = 10$ for PBM and CM models. The plotted curves correspond to the average over 200 independent sequences of recommendations (20 sequences per query). The (small) shaded areas depict the standard error of our regret estimates.

the neighborhood of the optimal recommendation. However, PB-MHB is computationally expensive (see Table 5.2) and lacks a theoretical analysis. Note also that GRAB suffers a high regret for $K = 10$ meaning that it does not find the right ranking among the selected items.

Finally, our algorithm UniRank and TopRank enjoy a logarithmic regret in all settings and UniRank has a lower regret than TopRank, except for CM model with $K = 5$ from iterations 10^5 to 10^7. However, TopRank is aware of the horizon T and may stop (over)exploring early, as can be observed in the CM model after iteration 10^5. If TopRank targets a horizon $T = 10^{12}$ or uses the doubling trick it suffers a higher regret than UniRank.

Regarding the computational complexity, as shown in Table 5.2, PB-MHB is significantly slower with a computation time per recommendation ten times higher than any other algorithm. These other algorithms have a similar computation time of approximately 1 ms per recommendation.

Overall, (i) only UniRank and TopRank are consistent over all settings, with a reasonable computation time, and a theoretical proof, and (ii) in 3 settings over 4 UniRank enjoys a smaller regret than TopRank even though UniRank does not require the knowledge of the horizon T.

Algorithm	Computation Time (ms)	
	K=5	K=10
UniRank	1.0 ± 0.2	1.2 ± 0.3
TopRank	0.7 ± 0.3	0.4 ± 0.1
PB-MHB	13.9 ± 4.9	20.8 ± 7.4
GRAB	0.9 ± 0.3	1.1 ± 0.4
CascadeKL-UCB	0.9 ± 0.0	1.2 ± 0.3

Table 5.2: Average computation time per recommendation. For each top 10 query of Yandex dataset, 20 runs are performed assuming CM model and $L = 10$.

5.5 Conclusion

We saw that this extension allows to tackle a wider range of click behavioral models which assume that positions and items have an impact on users' click and fall in the OLR setting. UniRank needs a different metric to find the best recommendation as CM may lead to multiple good recommendations although unimodality requires only a unique best possibility. UniRank leads to stable performance across settings.

Conclusion

Take away

Digital relationships with clients need to be nurtured by good recommendations. To provide such recommendations, two components are key:

— understand clients' click behavior since these clicks are a key interaction between clients and digital interfaces,

— adapt dynamically to clients' interactions by taking every clicks and every products consulted into account; This entails combining exploration and exploitation of possible recommendations in order to recommend more relevant items to clients.

These two aspects correspond to two research fields: the study of click behavioral models and the design of bandit algorithms. This thesis aims at adapting click behavioral models to the multiple-play semi-bandit setting while using as few assumptions as possible. Using bandits to tackle click behavioral models is equivalent to solving an online learning to rank problem (OLR) with clicks feedbacks defined by the number of available items L, the number of positions K and the set of distributions which gives to any list of recommendations the click probability on every couple (item, position). By adopting different angles, three contributions rise from the study of this problem.

The first angle adopted is a restriction of the OLR problem to position-based online learning to rank (PB-OLR). This restricted problem uses a set of distributions which gives the probability of clicking on an item displayed at a given position. This problem is particularly adapted to the full position-based model (PBM) which is a click behavioral model well-suited for our targeted industrial applications. The full PBM setting aims at recommending a ranking of K items among L according to their attractiveness and the visibility of the positions without any prior knowledge on these quantities. This full PBM setting has, to the best of our knowledge, rarely been tackled. Our proposed algorithms GRAB (parametric Graph for unimodal RAnking Bandit), PB-MHB (Position Based Metropolis-Hastings Bandit) and PB-LB (Position Based Langevin gradient Bandit) learn online both the user preferences and the gaze habits. Each algorithm adopts a different strategy to solve this problem.

GRAB uses a unimodal approach. To apply the unimodal bandit setting efficiently, we define a graph parameterized by a ranking on positions. This definition leads to a family of graphs where each graph maps all possible permutations of items according to the impact of each position. GRAB learns online the parameterization over this family of graphs to identify a unimodal graph. Then, GRAB builds the path to the best permutation of items in this unimodal graph and recommend the optimal list recommendation. We prove a regret upper-bound in $O(L/\Delta \log T)$ for this algorithm which reduces by a factor K^2 the bound which would be obtained without the unimodal setting. The strength of this algorithm derives first from the application of unimodality to OLR, and second from the family of graphs and its parameterization by the ranking of positions. Indeed, a naive application of unimodality leads to a bandit algorithm (denoted S-GRAB) exploring a graph a degree KL, which induces a regret of the same level as previous state of the art approaches: $O(LK/\Delta \log T)$. Then, by considering a set of graphs, GRAB handle graphs of degree L and hence reduces the regret by a factor K. On real and simulated data, GRAB quickly delivers good recommendations. Nevertheless, our second algorithm, PB-MHB, leads to better empirical regret.

The second approach, implemented in PB-MHB and PB-LB, consists in coupling Thompson Sampling bandits framework with an MCMC approximation to sample draws from the law induced by the full PBM. Indeed, this law is unusual and does not have any closed form usable in a Bayesian learning process. PB-MHB and PB-LB are two ways of applying MCMC approximations to this law. PBM has two sets of parameters, one for the probability of viewing each position, the other for the probability of the user to acknowledge each item to be relevant. As these two sets of PBM parameters have symmetrical roles in the full PBM, PB-MHB uses a Metropolis Hasting approximation with a Gibbs splitting to learn the values of the parameters of the PBM model. This leads to a more accurate recommendation with the lowest empirical regret achieved so far in the state-of-the-art with the same assumptions. Unfortunately, by treating each parameter and updating them separately, the computational time drastically increases. By applying a Langevin gradient descent, another MCMC approximation, PB-LB achieves a lower and more stable (according to the possible parameters) computation time and its regret performance is on part with state-of-the-art algorithms. Yet, the use of Langevin gradient implies a tedious tuning of the hyperparameters of the algorithm. For both algorithms, no theoretical analysis on the regret bound is given due to the complexity involved in the double randomness induced by the Thompson sampling framework and the MCMC

approximations.

This two contributions give strong results both theoretical and empirical to tackle the problem of list recommendation in the PBM setting.

The second angle of this thesis concerns the use of unimodality for list recommendations in more click behavioral models (than PBM). While the setting is more generic, again unimodality may lead to the design of efficient algorithms. Typically, a naive application of unimodality leads again to a bandit algorithm with a $O(LK/\Delta \log T)$ regret bound, which however remains at the level of state of the art algorithms. Unfortunately, GRAB cannot be adapted to general click behavioral models as it is strongly based on the assumption that the probability of clicking on an item i displayed at position k is independent of the items displayed displayed at other positions. Therefore, to tackle the initial OLR setting and still achieve a regret upper-bound in $O(L/\Delta \log T)$, UniRank (Unimodal Bandit Algorithm for Generic Online Ranking) applies unimodal bandits to a graph where nodes are ordered partitions of items instead of lists of items. This algorithm also uses pessimistic estimators instead of the estimators of the expected reward used by OSUB [13]. These two elements alter OSUB's exploration-exploitation strategy to reach a regret upper-bound in $O(L/\Delta \log T)$. Both GRAB and UniRank are original contributions in their use of unimodal bandit, and we believe that our theoretical analysis opens new perspectives for other semi-bandit settings. Experiments against state-of-the-art learning algorithms show that UniRank is stable in all settings. UniRank enjoys a smaller regret than GRAB in PBM and than CascadeKL-UCB [38] (the corresponding state-of-the-art) in CM. UniRank also has a much smaller computation time than PB-MHB.

To summarize, the initial problem tackled by this thesis was to dynamically adapt to users while respecting their click behavior. All three algorithms presented in this thesis answer this initial problem by tackling different aspects. In Table 5.3, the contributions are compared with respect to their theoretical and empirical results. GRAB gives theoretical guarantees for bandit recommendations according to PBM which is a simple yet relevant click behavioral model. PB-MHB and PB-LB give strong empirical results in this same setting with the addition of being computationally efficient for PB-LB. And UniRank provides a more generic approach to make bandit recommendations on a wider range of click behavioral models while conserving the same theoretical regret upper bound as GRAB.

Table 5.3: Comparison of the contributions of this thesis

Algorithm	Setting	Method	Theoretical analysis	Empirical regret*	Computation time**
PB-MHB	PB-OLR	Metropolis Hasting	unknown	2×10^3	17
PB-LB	PB-OLR	Langevin gradient	unknown	3×10^4	1
GRAB	PB-OLR	Unimodal	$O(L/\Delta \log T)$	1×10^4	0.8
UniRank	OLR	Unimodal	$O(L/\Delta \log T)$	1×10^4	1

* on Dataset Yandex L=10, K=5 at $T = 10^7$

** for one recommendation in ms

Contributions for Louis Vuitton

The work presented in this thesis is planned to be implemented. Indeed, evaluations on Louis Vuitton's customers' behavior are needed before using such algorithms online. The two initiatives described in this section aim at collecting data in a dynamic recommendation setting in order to proceed (in the future) to an evaluation on a corporate Louis Vuitton dataset.

As part of this thesis, an industrial application called "Dynamic Recommender" has been developed. The architecture of this application has been thought to be as flexible as possible in order to cover diverse needs such as product recommendations but also to test front pages on the website. To highlight the potential of using bandit-based dynamic learning at Vuitton, I developed a simple application where basic K-armed bandits such as Thompson Sampling or ϵ-greedy are available. Dynamic Recommender should ease the use of bandits to a few call to the application and manage its data automatically. The deployment of this application is currently in process. It should provide inspirational recommendations for Louis Vuitton's chatbot by March 2022. This project generates lots of interest from the business teams, which are implied in the co-creation of new types of use cases for Louis Vuitton. This team work feeds thoughts on the database management and the learning process, for example.

During this three-years thesis, I also supervised a six-month intern at Vuitton who worked on contextual bandits. The goal was to apply well-known contextual bandits to Louis Vuitton's products data, such as images or descriptions, and understand how these data should be integrated to a contextual bandit. This work aimed first at boosting

recommendation systems in production on Louis Vuitton website, and include contextual bandit to the application "Dynamic Recommender".

Echo chamber and exploration behavior

In the media, an *Echo chamber* is defined as a closed environment where each person's beliefs are amplified and reinforced by repetition and where he/she can only access information reflecting its own point of view. This insulates the person from opposite opinions and can lead to misinformation. This phenomenon can happen anywhere where information are exchanged, but on digital interfaces such as website and social medias, recommendation systems are designed to collect user interactions and provide precise recommendations according to these specific interactions. Recommendation systems primarily show users, contents that are similar to what they have already agreed on. Indeed, these contents are more likely to have a high probability to be pleasant to the user. This specific type of Echo Chamber is called a *filter bubble*. It raises many social concerns as it tends to wider the gap between social groups. Echo chambers prevent people to have complex and structured exchanges as they split information. Echo chambers isolate users from perspectives they have not expressed an interest in yet. In our context, which is e-commerce recommendations, filter bubbles can lead to user boredom.

The contributions presented in this thesis could lead to such pitfall if improperly applied. Unlike batch learning algorithms such as traditional collaborative filtering, bandit algorithms have an exploration mechanism. Exploration brings more heterogeneous recommendations in order to collect enough and varied information and to take consistent decisions. Nevertheless, bandit algorithms also have an optimisation goal. They aim at promoting items which have a high probability to satisfy users according to their past behavior. Thus, if you use a bandit algorithm on a homogeneous group, you will learn faster their expectation, but you will lock them up in a filter bubble. To prevent such behavior of algorithms, it is possible to apply aging laws to mitigate the importance of the best items and to reintroduce exploration. Another way would simply be to build recommendation use cases on heterogeneous groups in order to vary the recommendations. These solutions can be seen as business rules and do not change bandit algorithms objectives. To avoid such rules and risking a deterioration of the algorithm performances, bandit algorithms can include diversity in their objectives [53, 28, 52]. Rather than only maximizing click rates, a loss can be introduced to evaluate how similar products are within a list. This

solution can be seen as including a diversity constrain in a combinatorial bandit setting. Nevertheless, this constrain changes the setting presented in this thesis as it assumes that relevancy of a product does not depend only on the product itself but also on surrounding products.

Perspectives

This work is a contribution to Louis Vuitton's global project to enhance client online relationship with the brand. It opens multiple perspectives both in research and development on short or long terms. These perspectives can be divided into two categories. If there exists a strong methodology to address the subject and off-the-shelf algorithms to use in an industrial context then I will refer to it as *development subjects*. If there is no such methodology or existing algorithms and further work has to be done to transpose the subject into a research problem and bring a solution with strong guaranty to apply it then, I will refer to it as *research subjects*. I prioritize both types of subjects according to their relevancy for business and their immediate feasibility.

Short term research subjects One way to extend the contributions of this thesis is to transpose them to contextual bandits in order to speed up the learning process when using more items. The integration of unimodal bandit algorithms working on parametric spaces [14] may pave the way to efficient contextual recommendation systems handling larger sets of items and positions. Moreover, we would like to apply PB-MHB to environments where PBM parameters are evolving with time and where our learning setting could develop its full potential, especially when the parameter related to the position are evolving. Relevancy of items greatly changes over time. We could benefit in understanding this evolution in order to anticipate its impact on learning and adapt to it. Finally, in this work, only clicks have been taken into account to express clients' tastes. There are many other feedbacks which can be taken into account to characterize clients' needs and tastes, such as the actual buying of the product. These feedbacks show different levels of commitment of the client and have to be considered accordingly. For instance, a series of clicks may lead to a sell [64].

Short term development subjects As mentioned earlier, the "Dynamic Recommender" application has been developed to ease the use of bandit algorithms. Contextual

bandits could be added and used to enhance recommendation qualities on a large set of items. One of the main issue when industrializing bandit algorithms is to have an evaluation pipeline and to monitor performances. To solve this issue, many counterfactual estimators exist to measure offline the gain of new policies [49, 60] and should be implemented in order to monitor performances of new bandit algorithms but also to measure the shift in online recommendations. One should design properly the testing pipeline of bandit recommendations and measure the gain of such recommendations over other types of recommendation systems (a.k.a. A/B testing).

Long term research subjects To trust somebody who gives you an advice, he/she often needs to add an explanation. Another way to make recommendations convincing, is to build it through a conversation. These two aspects can be found in the literature associated to two fields: Explainable AI and Conversational AI. Explainable AI is nowadays associated with various degrees of explanations. For instance, when a recommendation is based on similar products, one can use the referenced products as an explanation. However, recommendation systems are becoming more and more complex with, for instance, neural recommendation systems [29], explainable AI addition will be needed to ensure trust between recommendation systems and users. This need for more explainable recommendations leads to various interesting topics such as constructing measures and datasets to evaluate recommendation explanations [45]. An other topic are counterfactual explanations for recommendation which study how users' actions change overtime [27, 63]. Moreover, adding explainability in such systems will enhance future performances as it can help spot mistakes. By adding understandable elements used by the algorithm to build its recommendations, human-in-the-loop scheme can be build around such algorithms. Humans will be able to understand and correct algorithms' decisions and algorithms will express new patterns of choice which will help sellers to perform better recommendations.

On the other hand, Conversationnal AI aims at building complex conversations by adapting to the user. This ability would help building trust and collect valuable information to help users. Users and systems can dynamically interact with conversationnal AI, enabling collecting explicit and immediate users' preferences. According to [20], this field raises many challenges among which building a mutli-turn conversational recommendation strategy which I found particularly interesting as it appears to me as a direct application of the exploration/exploitation trade-off on multiple levels: which number of questions to chose? which topics? which order? which items must be associated together? what type

of questions?

Another aspect that makes recommending a difficult task is the evolution of the needs and the behaviors of the users. An interesting path of research is to infer a user' motivation within a searching session [61, 30] to adequately make a recommendation through a minimal number of interactions and/or few information on him/her. A way to adapt to users needs is to use collective behaviour to understand how users interact with a recommendation. By using contextual bandits and users' first actions on a website, one can classify users according to their instant needs and use the collaborative and dynamic learning of bandits to recommend adequate item. Nevertheless identifying and characterizing users' instant needs, which are subjective and complex information is still difficult.

Long term development subjects Combinatorial bandits are a well known setting which can handle various situations thanks to constraints which can be put upon their set of arms. The main issue with this setting is that, depending on the constraints considered, the problem encountered can have a NP complexity and thus it has to be designed carefully. However, this class of bandits offers a possible solution to learn how to (aesthetically) match products. Indeed, the fashion business aims at selling products from different categories, for instance a hat and a bag. A way to do so is to recommend products that match well according to style, for example. Thus, being able to learn this type of constraints and applying them to recommend lists of items is relevant for fashion industries. Other methods exist based on neural networks and tensor completion to build looks from scratch or complete a scene with adequate items [8, 31].

BIBLIOGRAPHY

[1] Shipra Agrawal and Navin Goyal, « Near-Optimal Regret Bounds for Thompson Sampling », *in*: *Jour. of the ACM, JACM* 64.*5* (Sept. 2017), 30:1–30:24.

[2] Shipra Agrawal and Navin Goyal, « Thompson Sampling for Contextual Bandits with Linear Payoffs », *in*: *proc. of the 30th Int. Conf. on Machine Learning*, ICML'13, 2013.

[3] Maarten de Rijke Aleksandr Chuklin Ilya Markov, « Click Models for Web Search », *in*: *Click Models for Web Search*, ed. by Kamalika Chaudhuri and Masashi Sugiyama, http://clickmodels.weebly.com/the-book.html: Morgan and Claypool Publishers, 2015.

[4] P. Auer, N. Cesa-Bianchi, and P. Fischer, « Finite-time analysis of the multiarmed bandit problem », *in*: *Machine learning* 47.*2* (2002), pp. 235–256.

[5] Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer, « Finite-time Analysis of the Multiarmed Bandit Problem », *in*: *Machine Learning* 47.*2* (May 2002), pp. 235–256.

[6] Léon Bottou et al., « Counterfactual Reasoning and Learning Systems: The Example of Computational Advertising », *in*: *Journal of Machine Learning Research* 14.*65* (2013), pp. 3207–3260, URL: `http://jmlr.org/papers/v14/bottou13a.html`.

[7] Olivier Chapelle and Lihong Li, « An Empirical Evaluation of Thompson Sampling », *in*: *Advances in Neural Information Processing Systems 24*, NIPS'11, 2011.

[8] Huiyuan Chen et al., « Tops, Bottoms, and Shoes: Building Capsule Wardrobes via Cross-Attention Tensor Network », *in*: *Fifteenth ACM Conference on Recommender Systems*, New York, NY, USA: Association for Computing Machinery, 2021, pp. 453–462, ISBN: 9781450384582.

[9] Wei Chen, Yajun Wang, and Yang Yuan, « Combinatorial multi-armed bandit: General framework and applications », *in*: *proc. of the 30th Int. Conf. on Machine Learning*, ICML'13, 2013.

[10] Heng-Tze Cheng et al., « Wide & Deep Learning for Recommender Systems », *in*: *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*, DLRS 2016, Boston, MA, USA: Association for Computing Machinery, 2016, pp. 7–10, ISBN: 9781450347952, DOI: 10.1145/2988450.2988454, URL: https://doi.org/10.1145/2988450.2988454.

[11] Wang Chi Cheung, Vincent Tan, and Zixin Zhong, « A Thompson Sampling Algorithm for Cascading Bandits », *in*: *proc. of the 22nd Int. Conf. on Artificial Intelligence and Statistics*, AISTATS'19, 2019.

[12] Aleksandr Chuklin, Ilya Markov, and Maarten de Rijke, *Click Models for Web Search*, Morgan & Claypool Publishers, 2015.

[13] Richard Combes and Alexandre Proutière, « Unimodal bandits: Regret lower bounds and optimal algorithms », *in*: *proc. of the 31st Int. Conf. on Machine Learning, ICML'14*, 2014.

[14] Richard Combes, Alexandre Proutière, and Alexandre Fauquette, « Unimodal Bandits with Continuous Arms: Order-Optimal Regret without Smoothness », *in*: *Proc. ACM Meas. Anal. Comput. Syst.* 4.*1* (May 2020).

[15] Richard Combes et al., « Learning to Rank: Regret Lower Bounds and Efficient Algorithms », *in*: *proc. of the ACM SIGMETRICS Int. Conf. on Measurement and Modeling of Computer Systems*, 2015.

[16] Nick Craswell et al., « An Experimental Comparison of Click Position-bias Models », *in*: *proc. of the Int. Conf. on Web Search and Data Mining*, WSDM '08, 2008.

[17] Bianca Dumitrascu, Karen Feng, and Barbara Engelhardt, « PG-TS: Improved Thompson Sampling for Logistic Contextual Bandits », *in*: *Advances in Neural Information Processing Systems 31*, NIPS'18, 2018.

[18] Alain Durmus and Éric Moulines, « Nonasymptotic convergence analysis for the unadjusted Langevin algorithm », *in*: *The Annals of Applied Probability* 27.*3* (2017), pp. 1551–1587, ISSN: 10505164, URL: http://www.jstor.org/stable/26361410.

[19] Yi Gai, Bhaskar Krishnamachari, and Rahul Jain, « Combinatorial Network Optimization with Unknown Variables: Multi-Armed Bandits with Linear Rewards and Individual Observations », *in*: *IEEE/ACM Trans. Netw.* 20.*5* (Oct. 2012), pp. 1466–1478.

[20] Chongming Gao et al., « Advances and challenges in conversational recommender systems: A survey », *in*: *AI Open* 2 (2021), pp. 100–126, ISSN: 2666-6510, DOI: https://doi.org/10.1016/j.aiopen.2021.06.002, URL: https://www.sciencedirect.com/science/article/pii/S2666651021000164.

[21] Aurélien Garivier and Olivier Cappé, « The KL-UCB Algorithm for Bounded Stochastic Bandits and Beyond », *in*: *proc. of the 24th Annual Conf. on Learning Theory*, COLT'11, 2011.

[22] Aurélien Garivier and Olivier Cappé, *The KL-UCB Algorithm for Bounded Stochastic Bandits and Beyond*, 2013, arXiv: 1102.2490 [math.ST].

[23] Camille-Sovanneary Gauthier, Romaric Gaudel, and Élisa Fromont, « Bandit Algorithm for both Unknown Best Position and Best Item Display on Web Pages », *in*: *19th International Symposium on Intelligent Data Analysis, IDA*, vol. 12695, Lecture Notes in Computer Science, Springer, 2021, pp. 209–221.

[24] Camille-Sovanneary Gauthier, Romaric Gaudel, and Élisa Fromont, « Bandits manchots avec échantillonnage de Thompson pour des recommandations multiples suivant un modèle fondé sur les positions », *in*: *Joint Conferences CAp and RFIAP 2020, CAp + RFIAP*, 2020.

[25] Camille-Sovanneary Gauthier, Romaric Gaudel, and Élisa Fromont, « Ordonnancement d'objets par bandits unimodaux sur des graphes paramétriques », *in*: *Conférence sur l'Apprentissage automatique, CAp 21*, 2021.

[26] Camille-Sovanneary Gauthier et al., « Parametric Graph for Unimodal Ranking Bandit », *in*: *Proceedings of the 38th International Conference on Machine Learning, ICML*, vol. 139, Proceedings of Machine Learning Research, PMLR, 2021, pp. 3630–3639.

[27] Azin Ghazimatin et al., « PRINCE: Provider-side Interpretability with Counterfactual Explanations in Recommender Systems », *in*: *Proceedings of the 13th International Conference on Web Search and Data Mining* (Jan. 2020), DOI: 10.1145/3336191.3371824.

[28] Hédi Hadiji et al., *Diversity-Preserving K-Armed Bandits, Revisited*, 2020, arXiv: 2010.01874 [stat.ML].

[29] Xiangnan He et al., « Neural Collaborative Filtering », *in*: *Proceedings of the 26th International Conference on World Wide Web*, WWW '17, Perth, Australia: International World Wide Web Conferences Steering Committee, 2017, pp. 173–182, ISBN: 9781450349130, DOI: 10.1145/3038912.3052569, URL: https://doi.org/10.1145/3038912.3052569.

[30] Jyun Yu Jiang et al., « Learning to represent human motives for goal-directed web browsing », English (US), *in*: *RecSys 2021 - 15th ACM Conference on Recommender Systems*, RecSys 2021 - 15th ACM Conference on Recommender Systems, Publisher Copyright: © 2021 ACM.; 15th ACM Conference on Recommender Systems, RecSys 2021 ; Conference date: 27-09-2021 Through 01-10-2021, Association for Computing Machinery, Inc, Sept. 2021, pp. 361–371, DOI: 10.1145/3460231.3474260.

[31] Wang-Cheng Kang et al., « Complete the Look: Scene-Based Complementary Product Recommendation », *in*: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019), pp. 10524–10533.

[32] Sumeet Katariya et al., « Bernoulli Rank-1 Bandits for Click Feedback », *in*: *proc. of the 26th International Joint Conference on Artificial Intelligence*, IJCAI'17, 2017.

[33] Sumeet Katariya et al., « DCM Bandits: Learning to Rank with Multiple Clicks », *in*: *proc. of the 33rd Int. Conf. on Machine Learning*, ICML'16, 2016.

[34] Sumeet Katariya et al., « Stochastic Rank-1 Bandits », *in*: *proc. of the 20th Int. Conf. on Artificial Intelligence and Statistics*, AISTATS'17, 2017.

[35] Jaya Kawale et al., « Efficient Thompson Sampling for Online Matrix Factorization Recommendation », *in*: *Advances in Neural Information Processing Systems 28*, NIPS'15, 2015.

[36] Junpei Komiyama, Junya Honda, and Hiroshi Nakagawa, « Optimal Regret Analysis of Thompson Sampling in Stochastic Multi-armed Bandit Problem with Multiple Plays », *in*: *proc. of the 32nd Int. Conf. on Machine Learning*, ICML'15, 2015.

[37] Junpei Komiyama, Junya Honda, and Akiko Takeda, « Position-based Multiple-play Bandit Problem with Unknown Position Bias », *in*: *Advances in Neural Information Processing Systems 30*, NIPS'17, 2017.

[38] Branislav Kveton et al., « Cascading Bandits: Learning to Rank in the Cascade Model », *in*: *proc. of the 32nd Int. Conf. on Machine Learning*, ICML'15, Lille, France, 2015.

[39] Branislav Kveton et al., « Combinatorial Cascading Bandits », *in*: *Advances in Neural Information Processing Systems 28*, NIPS'15, 2015.

[40] Branislav Kveton et al., « Tight Regret Bounds for Stochastic Combinatorial Semi-Bandits », *in*: *proc. of the 18th Int. Conf. on Artificial Intelligence and Statistics*, AISTATS'15, 2015.

[41] Paul Lagrée, Claire Vernade, and Olivier Cappé, « Multiple-play Bandits in the Position-based Model », *in*: *Advances in Neural Information Processing Systems 30*, NIPS'16, 2016.

[42] John Langford, Alexander Strehl, and Jennifer Wortman, « Exploration Scavenging », *in*: ICML '08, Helsinki, Finland: Association for Computing Machinery, 2008, pp. 528–535, ISBN: 9781605582054, DOI: 10.1145/1390156.1390223, URL: https://doi.org/10.1145/1390156.1390223.

[43] Tor Lattimore et al., « TopRank: A practical algorithm for online stochastic ranking », *in*: *Advances in Neural Information Processing Systems 31*, NIPS'18, 2018.

[44] Chang Li et al., « BubbleRank: Safe Online Learning to Re-Rank via Implicit Click Feedback », *in*: *proc. of the 35th Uncertainty in Artificial Intelligence Conference*, UAI'19, 2019.

[45] Lei Li, Yongfeng Zhang, and Li Chen, « EXTRA: Explanation Ranking Datasets for Explainable Recommendation », *in*: *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, New York, NY, USA: Association for Computing Machinery, 2021, pp. 2463–2469, ISBN: 9781450380379.

[46] Shuai Li et al., « Contextual Combinatorial Cascading Bandits », *in*: *proc. of the 33rd Int. Conf. on Machine Learning*, ICML'16, 2016.

[47] Pasquale Lops, Marco Degemmis, and Giovanni Semeraro, « Content-based Recommender Systems: State of the Art and Trends », *in*: *Recommender Systems Handbook*, 2011.

[48] Eric Mazumdar et al., « On Thompson Sampling with Langevin Algorithms », *in*: *proc. of the 37th Int. Conf. on Machine Learning*, ICML'20, 2020.

[49] Yusuke Narita, Shota Yasui, and Kohei Yata, *Efficient Counterfactual Learning from Bandit Feedback*, 2018, arXiv: 1809.03084 [cs.LG].

[50] Radford M. Neal, *Probabilistic Inference Using Markov Chain Monte Carlo Methods*, tech. rep., University of Zurich, Department of Informatics, Sept. 1993.

[51] Olivier Nicol, Jérémie Mary, and Philippe Preux, *Improving offline evaluation of contextual bandit algorithms via bootstrapping techniques*, 2014, arXiv: 1405.3536 [stat.ML].

[52] Javier Parapar and Filip Radlinski, « Diverse User Preference Elicitation with Multi-Armed Bandits », *in*: *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, WSDM '21, Virtual Event, Israel: Association for Computing Machinery, 2021, pp. 130–138, ISBN: 9781450382977, DOI: 10.1145/3437963.3441786.

[53] Filip Radlinski, Robert Kleinberg, and Joachims Thorsten, « Learning diverse rankings with multi-armed bandits », *in*: *proc. of the 25th Int. Conf. on Machine Learning*, ICML'08, 2008.

[54] Lyle Ramshaw and Robert E. Tarjan, *On Minimum-Cost Assignments in Unbalanced Bipartite Graphs*, tech. rep., HP research labs, 2012.

[55] Matthew Richardson, Ewa Dominowska, and Robert Ragno, « Predicting Clicks: Estimating the Click-Through Rate for New Ads », *in*: *proc. of the 16th International World Wide Web Conference*, WWW '07, 2007.

[56] Carlos Riquelme, George Tucker, and Jasper Snoek, « Deep Bayesian Bandits Showdown: An Empirical Comparison of Bayesian Deep Networks for Thompson Sampling », *in*: *proc. of the Int. Conf. on Learning Representations*, ICLR'18, 2018.

[57] J. Ben Schafer et al., « Collaborative Filtering Recommender Systems », *in*: *The Adaptive Web: Methods and Strategies of Web Personalization*, Berlin, Heidelberg: Springer-Verlag, 2007, pp. 291–324, ISBN: 9783540720782.

[58] Flore Sentenac et al., « Pure Exploration and Regret Minimization in Matching Bandits », *in*: *Proc. of the 38th Int. Conf. on Machine Learning*, ICML'21, 2021, pp. 9434–9442.

[59] Adith Swaminathan and Thorsten Joachims, *Counterfactual Risk Minimization: Learning from Logged Bandit Feedback*, 2015, arXiv: 1502.02362 [cs.LG].

[60] Adith Swaminathan et al., « Off-policy evaluation for slate recommendation », *in*: *Advances in Neural Information Processing Systems*, ed. by I. Guyon et al., vol. 30, Curran Associates, Inc., 2017, URL: https://proceedings.neurips.cc/paper/2017/file/5352696a9ca3397beb79f116f3a33991-Paper.pdf.

[61] Jacopo Tagliabue et al., *SIGIR 2021 E-Commerce Workshop Data Challenge*, 2021, arXiv: 2104.09423 [cs.IR].

[62] William R Thompson, « On The Likelihood that one Unknown Probability Exceeds Another in View of the Evidence of two Samples », *in*: *Biometrika* 25.*3-4* (Dec. 1933), pp. 285–294, ISSN: 0006-3444, DOI: 10.1093/biomet/25.3-4.285, URL: https://doi.org/10.1093/biomet/25.3-4.285.

[63] Khanh Hiep Tran, Azin Ghazimatin, and Rishiraj Saha Roy, « Counterfactual Explanations for Neural Recommenders », *in*: *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval* (July 2021), DOI: 10.1145/3404835.3463005.

[64] Claire Vernade, Andras Gyorgy, and Timothy Mann, *Non-Stationary Delayed Bandits with Intermediate Observations*, 2020, arXiv: 2006.02119 [stat.ML].

[65] Fuzheng Zhang et al., « Collaborative Knowledge Base Embedding for Recommender Systems », *in*: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, San Francisco, California, USA: Association for Computing Machinery, 2016, pp. 353–362, ISBN: 9781450342322, DOI: 10.1145/2939672.2939673, URL: https://doi.org/10.1145/2939672.2939673.

[66] Masrour Zoghi et al., « Online Learning to Rank in Stochastic Click Models », *in*: *proc. of the 34th Int. Conf. on Machine Learning*, ICML'17, 2017.

[67] Shi Zong et al., « Cascading Bandits for Large-scale Recommendation Problems », *in*: *proc. of the 32nd Conference on Uncertainty in Artificial Intelligence*, UAI '16, 2016.

APPENDIX

Appendix A

GRAB

The appendix is organized as follows. We first list most of the notations used in the paper in Appendix A.1. Lemma 1 is proved in Appendix A.2. In Appendix A.3, we recall a Lemma from [13] used by our own Lemmas and Theorems, and then in Appendices A.4 to A.6 we respectively prove Theorem 2, Lemma 2, and Lemma 3. Finally in Appendix A.7 we introduce and discuss S-GRAB.

A.1 Notations

The following table summarize the notations used through the paper and the appendix.

Symbol	Meaning
T	Time horizon
t	iteration
L	number of items
i	index of an item
K	number of positions in a recommendation
k	index of a position
$[n]$	set of integers $\{1, \dots, n\}$
$\mathcal{P}_K^L$	set of permutations of K distinct items among L
$\boldsymbol{\theta}$	vectors of probabilities of click
θ_i	probability of click on item i
$\boldsymbol{\kappa}$	vectors of probabilities of view
κ_k	probability of view at position k
$\mathcal{A}$	set of bandit arms
$\boldsymbol{a}$	an arm in $\mathcal{A}$
$\boldsymbol{a}(t)$	the arm chosen at iteration t
$\tilde{\boldsymbol{a}}(t)$	best arm at iteration t given the previous choices and feedbacks (called leader)

Continued on next page

Symbol	Meaning
$\boldsymbol{a}^*$	BEST ARM
G	GRAPH CARRYING A PARTIAL ORDER ON $\mathcal{A}$
γ	MAXIMUM DEGREE OF G
$\mathcal{N}_G(\tilde{\boldsymbol{a}}(t))$	NEIGHBORHOOD OF $\tilde{a}(t)$ GIVEN G
$\rho_{i,k}$	PROBABILITY OF CLICK ON ITEM i DISPLAYED AT POSITION k
$\boldsymbol{c}(t)$	CLICKS VECTOR AT ITERATION t
$r(t)$	REWARD COLLECTED AT ITERATION t, $r(t) = \sum_{k=1}^{K} c_k(t)$
$\mu_{\boldsymbol{a}}$	EXPECTATION OF $r(t)$ WHILE RECOMMENDING $\boldsymbol{a}$, $\mu_{\boldsymbol{a}} = \sum_{k=1}^{K} \rho_{a_k,k}$
μ^*	HIGHEST EXPECTED REWARD, $\mu^* = \max_{\boldsymbol{a} \in \mathcal{P}_K^L} \mu_{\boldsymbol{a}}$
Δ_a	GAP BETWEEN μ_a AND μ^*
Δ_{min}	MINIMAL VALUE FOR Δ_a
Δ	GENERIC REWARD GAP BETWEEN ONE OF THE SUB-OPTIMAL ARMS AND ONE OF THE BEST ARMS
$R(T)$	CUMULATIVE (PSEUDO-)REGRET, $R(T) = T\mu^* - \mathbb{E}\left[\sum_{t=1}^{T} \mu_{\boldsymbol{a}(t)}\right]$
$\Pi_{\boldsymbol{\rho}}(\boldsymbol{a})$	SET OF PERMUTATIONS IN $\mathcal{P}_K^K$ ORDERING THE POSITIONS S.T. $\rho_{a_{\pi_1},\pi_1} \geqslant \rho_{a_{\pi_2},\pi_2} \geqslant \cdots \geqslant \rho_{a_{\pi_K},\pi_K}$
$\boldsymbol{\pi}$	ELEMENT OF $\Pi_{\boldsymbol{\rho}}(\boldsymbol{a})$
$\tilde{\boldsymbol{\pi}}$	ESTIMATION OF $\boldsymbol{\pi}$
$\boldsymbol{a} \circ (\pi_k, \pi_{k+1})$	PERMUTATION SWAPPING ITEMS IN POSITIONS π_k AND π_{k+1}
$\boldsymbol{a}[\pi_K := i]$	PERMUTATION LEAVING $\boldsymbol{a}$ THE SAME FOR ANY POSITION EXCEPT π_K FOR WHICH $\boldsymbol{a}[\pi_K := i]_{\pi_K} = i$
$\mathcal{F}$	RANKINGS OF POSITIONS RESPECTING $\Pi_{\boldsymbol{\rho}}$, $\mathcal{F} = (\boldsymbol{\pi}_{\boldsymbol{a}})_{\boldsymbol{a} \in \mathcal{P}_K^L}$ S.T. $\forall \boldsymbol{a} \in \mathcal{P}_K^L, \boldsymbol{\pi}_{\boldsymbol{a}} \in \Pi_{\boldsymbol{\rho}}(\boldsymbol{a})$
$T_{i,k}(t)$	NUMBER OF ITERATIONS S.T. ITEM i HAS BEEN DISPLAYED AT POSITION k, $T_{i,k}(t) = \sum_{s=1}^{t-1} \mathbb{1}\{a_k(s) = i\}$
$\tilde{T}_{\boldsymbol{a}}(t)$	NUMBER OF ITERATIONS S.T. THE LEADER WAS $\boldsymbol{a}$, $\tilde{T}_{\boldsymbol{a}}(t) \overset{def}{=} \sum_{s=1}^{t-1} \mathbb{1}\{\tilde{\boldsymbol{a}}(s) = \boldsymbol{a}\}$
$T_{\boldsymbol{a}}(t)$	NUMBER OF ITERATIONS S.T. THE CHOSEN ARM WAS $\boldsymbol{a}$, $T_{\boldsymbol{a}}(t) = \sum_{s=1}^{t-1} \mathbb{1}\{\boldsymbol{a}(s) = \boldsymbol{a}\}$
$T_{\boldsymbol{a}}^{\tilde{\boldsymbol{a}}}(t)$	NUMBER OF ITERATIONS S.T. THE LEADER WAS $\tilde{\boldsymbol{a}}$, THE CHOSEN ARM WAS $\boldsymbol{a}$, AND $\boldsymbol{a}$ WAS CHOSEN BY THE ARGMAX ON $\sum_{k=1}^{K} b_{a_k,k}(t)$: $T_{\boldsymbol{a}}^{\tilde{\boldsymbol{a}}}(t) = \sum_{s=1}^{t-1} \mathbb{1}\left\{\tilde{\boldsymbol{a}}(s) = \tilde{\boldsymbol{a}}, \boldsymbol{a}(s) = \boldsymbol{a}, \tilde{T}_{\tilde{\boldsymbol{a}}}(s)/L \notin \mathbb{N}\right\}$
$\hat{\rho}_{i,k}(t)$	ESTIMATION OF $\rho_{i,k}$ AT ITERATION t, $\hat{\rho}_{i,k}(t) = \frac{1}{T_{i,k}(t)} \sum_{s=1}^{t-1} \mathbb{1}\{a_k(s) = i\} c_k(s)$

CONTINUED ON NEXT PAGE

Symbol	Meaning
$b_{i,k}(t)$	Kullback-Leibler index of $\hat{\rho}_{i,k}(t)$, $b_{i,k}(t) = f\left(\hat{\rho}_{i,k}(t), T_{i,k}(t), \tilde{T}_{\tilde{\boldsymbol{a}}(t)}(t)+1\right)$
f	Kullback-Leibler index function, $f(\hat{\rho}, s, t) = \sup\{p \in [\hat{\rho}, 1] : s \times \mathrm{kl}(\hat{\rho}, p) \leq \log(t) + 3\log(\log(t))\}$,
$\mathrm{kl}(p, q)$	Kullback-Leibler divergence from a Bernoulli distribution of mean p to a Bernoulli distribution of mean q, $\mathrm{kl}(p,q) = p\log\left(\frac{p}{q}\right) + (1-p)\log\left(\frac{1-p}{1-q}\right)$
$B_{\boldsymbol{a}}(t)$	pseudo-sum of indices of $\boldsymbol{a}$ at iteration t, $B_{\boldsymbol{a}}(t) = \sum_{k=1}^{K} b_{a_k,k}(t) - \sum_{k=1}^{K} b_{\tilde{a}_k(t),k}(t)$
$\mathcal{N}_{\pi^*}(a^*)$	neighborhood of the best arm
$K_{\boldsymbol{a}}$	(with combinatorial bandit setting) number of elements in $\boldsymbol{a}$ but not in $\boldsymbol{a}^*$, $K_{\boldsymbol{a}} = \min_{\boldsymbol{a}^* \in \mathcal{A}: \mu_{\boldsymbol{a}^*} = \mu^*} \lvert \boldsymbol{a} \setminus \boldsymbol{a}^* \rvert$
K_{max}	(with combinatorial bandit setting) maximal number of elements in a sub-optimal arm $\boldsymbol{a}$ but not in an optimal arm a^*, $K_{max} = \max_{\boldsymbol{a} \in \mathcal{A}: \mu_{\boldsymbol{a}} \neq \mu^*} K_{\boldsymbol{a}}$
$c^*(\boldsymbol{\theta}, \boldsymbol{\kappa})$	coefficient in the regret bound of PMED
c	(in ε_n-greedy) parameter controlling the probability of exploration
c	(in PB-MHB) parameter controlling size of the step in the Metropolis Hasting inference
m	(in PB-MHB) number of step in the Metropolis Hasting inference

Table A.1: Summary of the notations of Chapter 3.

A.2 Proof of Lemma 1 (PBM Fulfills Assumption 1)

Proof of Lemma 1. Let $(L, K, (\rho_{i,k})_{(i,k)\in[L]\times[K]})$ be an online learning to rank (OLR) problem with users following PBM, with positive probabilities of looking at a given position. Therefore, there exists $\boldsymbol{\theta} \in [0,1]^L$ and $\boldsymbol{\kappa} \in (0,1]^K$ such that for any item i and any position k, $\rho_{i,k} = \theta_i \kappa_k$.

Let $\boldsymbol{a} \in \mathcal{P}_K^L$ be a recommendation, and let $\boldsymbol{\pi} \in \Pi_{\boldsymbol{\rho}}(\boldsymbol{a})$ be an appropriate ranking of

positions. One of the four following properties is satisfied:

$$\exists k \in [K-1] \text{ s.t. } \theta_{a_{\pi_k}} < \theta_{a_{\pi_{k+1}}}, \tag{A.1}$$

$$\exists k \in [K-1] \text{ s.t. } \kappa_{\pi_k} < \kappa_{\pi_{k+1}}, \tag{A.2}$$

$$\exists i \in [L] \setminus \boldsymbol{a}([K]) \text{ s.t. } \theta_{a_{\pi_K}} < \theta_i, \tag{A.3}$$

$$\begin{cases} \forall k \in [K-1], \theta_{a_{\pi_k}} \geqslant \theta_{a_{\pi_{k+1}}} \\ \forall k \in [K-1], \kappa_{\pi_k} \geqslant \kappa_{\pi_{k+1}} \\ \forall i \in [L] \setminus \boldsymbol{a}([K]), \theta_{a_{\pi_K}} \geqslant \theta_i \end{cases}. \tag{A.4}$$

Let prove, by considering each of these properties one by one, that $\boldsymbol{a}$ is either one of the best arms, or $\boldsymbol{a}$ fulfills either Property (2) or Property (3) of Assumption 1.

If Property (A.1) is satisfied and $\theta_{a_{\pi_k}} = 0$, then by definition of $\boldsymbol{\pi}$ and $\Pi_{\boldsymbol{\rho}}(\boldsymbol{a})$, $0 = \theta_{a_{\pi_k}} \kappa_{\pi_k} \geqslant \theta_{a_{\pi_{k+1}}} \kappa_{\pi_{k+1}} > 0$ which is absurd.

Therefore, If Property (A.1) is satisfied, $\frac{\theta_{a_{\pi_{k+1}}}}{\theta_{a_{\pi_k}}} > 1$.

Note that by definition of $\boldsymbol{\pi}$ and $\Pi_{\boldsymbol{\rho}}(\boldsymbol{a})$, and as $\rho_{i,k} = \theta_i \kappa_k$, $\theta_{a_{\pi_k}} \kappa_{\pi_k} \geqslant \theta_{a_{\pi_{k+1}}} \kappa_{\pi_{k+1}}$.

Hence $\kappa_{\pi_k} \geqslant \frac{\theta_{a_{\pi_{k+1}}}}{\theta_{a_{\pi_k}}} \kappa_{\pi_{k+1}} > \kappa_{\pi_{k+1}}$, and

$$\begin{aligned} \mu_{\boldsymbol{a}} - \mu_{\boldsymbol{a} \circ (\pi_k, \pi_{k+1})} &= \theta_{a_{\pi_k}} \kappa_{\pi_k} + \theta_{a_{\pi_{k+1}}} \kappa_{\pi_{k+1}} - \left(\theta_{a_{\pi_{k+1}}} \kappa_{\pi_k} + \theta_{a_{\pi_k}} \kappa_{\pi_{k+1}} \right) \\ &= \left(\theta_{a_{\pi_k}} - \theta_{a_{\pi_{k+1}}} \right) \left(\kappa_{\pi_k} - \kappa_{\pi_{k+1}} \right) \\ &< 0, \end{aligned}$$

meaning $\mu_{\boldsymbol{a}} < \mu_{\boldsymbol{a} \circ (\pi_k, \pi_{k+1})}$, which corresponds to Property (2) of Assumption 1.

Similarly, if Property (A.2) is satisfied, then Property (2) of Assumption 1 is fulfilled.

If Property (A.3) is satisfied,

$$\begin{aligned} \mu_{\boldsymbol{a}} - \mu_{\boldsymbol{a}[\pi_K := i]} &= \theta_{a_{\pi_K}} \kappa_{\pi_K} - \theta_i \kappa_{\pi_K} \\ &= \left(\theta_{a_{\pi_K}} - \theta_i \right) \kappa_{\pi_K} \\ &< 0. \end{aligned}$$

Hence $\mu_{\boldsymbol{a}} < \mu_{\boldsymbol{a}[\pi_K := i]}$, which corresponds to Property (3) of Assumption 1.

Finally, if Property (A.4) is satisfied, $\mu_{\boldsymbol{a}} = \mu^*$.

Overall, either $\boldsymbol{a}$ is one of the best arms, or $\boldsymbol{a}$ fulfills Property (2) of Assumption 1, or $\boldsymbol{a}$ fulfills Property (3) of Assumption 1, which concludes the proof.

□

A.3 Preliminary to the Analysis of GRAB

The analysis of GRAB requires a control of the number of high deviations, as expressed by Lemma $B.1$ of [13]. Let us recall this lemma, which we denote Lemma 6 in current paper.

Lemma 6 (Lemma B.1 of [13])**.** *Let $i \in [L]$, $k \in [K]$, $\epsilon > 0$. Define $\mathcal{F}(T)$ the σ-algebra generated by $(\boldsymbol{c}(t))_{t\in[T]}$. Let $\Lambda \subseteq \mathbb{N}$ be a random set of instants. Assume that there exists a sequence of random sets $(\Lambda(s))_{s\geq 1}$ such that (i) $\Lambda \subseteq \bigcup_{s\geq 1}\Lambda(s)$, (ii) for all $s \geqslant 1$ and all $t \in \Lambda(s)$, $T_{i,k}(t) \geq \epsilon s$, (iii) $|\Lambda(s)| \leqslant 1$, and (iv) the event $t \in \Lambda(s)$ is $\mathcal{F}_t$-measurable. Then for all $\delta > 0$,*

$$\mathbb{E}\left[\sum_{t\geq 1} \mathbb{1}\{t \in \Lambda, |\hat{\rho}_{i,k}(t) - \rho_{i,k}| \geqslant \delta\}\right] \leqslant \frac{1}{\epsilon\delta^2}$$

A.4 Proof of Theorem 2 (Upper-bound on the Regret of KL-CombUCB)

Proof of Theorem 2. Let $\boldsymbol{a} \in \mathcal{A}$ be a sub-optimal arm. Let $\boldsymbol{a}^* \in \mathcal{A}$ be an optimal arm such that $|\boldsymbol{a} \setminus \boldsymbol{a}^*| = K_{\boldsymbol{a}}$.

We denote $\bar{K_{\boldsymbol{a}}} \overset{def}{=} |\boldsymbol{a}^* \setminus \boldsymbol{a}|$, $T_{\boldsymbol{a}}(t) \overset{def}{=} \sum_{s=1}^{t-1} \mathbb{1}\{\boldsymbol{a}(s) = \boldsymbol{a}\}$ the number of time the arm $\boldsymbol{a}$ has been drawn, and $T_e(t) \overset{def}{=} \sum_{s=1}^{t-1} \mathbb{1}\{e \in \boldsymbol{a}(s)\}$ the number of time the element e was in the drawn arm.

Let decompose the expected number of iterations at which the permutation $\boldsymbol{a}$ is recommended:

$$\begin{aligned}
\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{\boldsymbol{a}(t) = \boldsymbol{a}\}\right] \leqslant & \sum_{e\in\boldsymbol{a}\setminus\boldsymbol{a}^*} \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\left\{\boldsymbol{a}(t) = \boldsymbol{a}, |\hat{\rho}_e(t) - \rho_e| \geqslant \frac{\Delta_{\boldsymbol{a}}}{2K_{\boldsymbol{a}}}\right\}\right] \\
& + \sum_{e\in\boldsymbol{a}^*\setminus\boldsymbol{a}} \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{b_e(t) \leqslant \rho_e\}\right] \\
& + \mathbb{E}\left[\sum_{t=|E|}^{T} \mathbb{1}\left\{\boldsymbol{a}(t) = \boldsymbol{a}, \forall e \in \boldsymbol{a} \setminus \boldsymbol{a}^*, |\hat{\rho}_e(t) - \rho_e| < \frac{\Delta_{\boldsymbol{a}}}{2K_{\boldsymbol{a}}}, \forall e \in \boldsymbol{a}^* \setminus \boldsymbol{a}, b_e(t) > \rho_e\right\}\right] \\
& + |E|.
\end{aligned}$$

The proof consists in upper-bounding each term on the right-hand side.

First Term Let $e \in \boldsymbol{a} \setminus \boldsymbol{a}^*$, and denote $A_e = \left\{t \in [T] : \boldsymbol{a}(t) = \boldsymbol{a}, |\hat{\rho}_e(t) - \rho_e| \geqslant \frac{\Delta_{\boldsymbol{a}}}{2K_{\boldsymbol{a}}}\right\}$.

$A_e \subseteq \bigcup_{s\in\mathbb{N}} \Lambda_k(s)$, where $\Lambda_k(s) \overset{def}{=} \{t \in A_e : T_{\boldsymbol{a}}(t) = s\}$. For any integer value s, $|\Lambda_k(s)| \leqslant 1$ as $T_{\boldsymbol{a}}(t)$ increases for each $t \in A_e$. Note that for each $s \in \mathbb{N}$ and $n \in \Lambda_k(s)$, $T_e(n) \geqslant T_{\boldsymbol{a}}(n) = s$. Then, by Lemma 6

$$\begin{aligned}\mathbb{E}\left[|A_e|\right] &\leq \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{t \in A_e\}\right] \\ &= \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\left\{t \in A_e, |\hat{\rho}_e(t) - \rho_e| \geqslant \frac{\Delta_{\boldsymbol{a}}}{2K_{\boldsymbol{a}}}\right\}\right] \\ &\leqslant \frac{4K_{\boldsymbol{a}}^2}{\Delta_{\boldsymbol{a}}^2}.\end{aligned}$$

Hence, $\sum_{e\in\boldsymbol{a}\setminus\boldsymbol{a}^*} \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\left\{\boldsymbol{a}(t) = \boldsymbol{a}, |\hat{\rho}_e(t) - \rho_e| \geqslant \frac{\Delta_{\boldsymbol{a}}}{2K_{\boldsymbol{a}}}\right\}\right] = \sum_{e\in\boldsymbol{a}\setminus\boldsymbol{a}^*} \mathbb{E}\left[|A_e|\right] \leqslant \frac{4K_{\boldsymbol{a}}^3}{\Delta_{\boldsymbol{a}}^2}$.

Second Term Let $e \in \boldsymbol{a}^* \setminus \boldsymbol{a}$, and denote $B_e \overset{def}{=} \{t \in [T] : b_e(t) \leqslant \rho_e\}$.

By Theorem 10 of [21], $\mathbb{E}\left[|B_e|\right] = O(\log\log T)$, so $\sum_{e\in\boldsymbol{a}^*\setminus\boldsymbol{a}} \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{b_e(t) \leqslant \rho_e\}\right] = \mathcal{O}(\bar{K}_{\boldsymbol{a}} \log\log T)$.

Third Term Let note

$$C \overset{def}{=} \left\{t \in [T] \setminus [|E|] : \boldsymbol{a}(t) = \boldsymbol{a}, \forall e \in \boldsymbol{a} \setminus \boldsymbol{a}^*, |\hat{\rho}_e(t) - \rho_e| < \frac{\Delta_{\boldsymbol{a}}}{2K_{\boldsymbol{a}}}, \forall e \in \boldsymbol{a}^* \setminus \boldsymbol{a}, b_e(t) > \rho_e\right\}.$$

Let $t \in C$.

At each step of the initialization phase, the algorithm removes at least one element e of the set $\tilde{E}$ of unseen elements. Therefore, the initialization lasts at most $|E|$ iterations. Hence, at iteration t, $\boldsymbol{a}(t) = \boldsymbol{a}$ is chosen as $\sum_{e\in\boldsymbol{a}} b_e(t) = \max_{\boldsymbol{a}'\in\mathcal{A}} \sum_{e\in\boldsymbol{a}'} b_e(t)$.

Then, by Pinsker's inequality and the fact that $t \leqslant T$, and $T_e(t) \geqslant T_{\boldsymbol{a}}(t)$ for any e in

$\boldsymbol{a}$,

$$
\begin{aligned}
0 &\leqslant \sum_{e\in\boldsymbol{a}} b_e(t) - \sum_{e\in\boldsymbol{a}^*} b_e(t) \\
&= \sum_{e\in\boldsymbol{a}\setminus\boldsymbol{a}^*} b_e(t) - \sum_{e\in\boldsymbol{a}^*\setminus\boldsymbol{a}} b_e(t) \\
&\leqslant \sum_{e\in\boldsymbol{a}\setminus\boldsymbol{a}^*} \hat{\rho}_e(t) + \sqrt{\frac{\log(t)+3\log(\log(t))}{2T_e(t)}} - \sum_{e\in\boldsymbol{a}^*\setminus\boldsymbol{a}} b_e(t) \\
&< \sum_{e\in\boldsymbol{a}\setminus\boldsymbol{a}^*} \rho_e + \frac{\Delta_{\boldsymbol{a}}}{2K_{\boldsymbol{a}}} + \sqrt{\frac{\log(T)+3\log(\log(T))}{2T_{\boldsymbol{a}}(t)}} - \sum_{e\in\boldsymbol{a}^*\setminus\boldsymbol{a}} \rho_e \\
&\leqslant \sum_{e\in\boldsymbol{a}} \rho_e - \sum_{e\in\boldsymbol{a}^*} \rho_e + K_{\boldsymbol{a}}\frac{\Delta_{\boldsymbol{a}}}{2K_{\boldsymbol{a}}} + K_{\boldsymbol{a}}\sqrt{\frac{\log(T)+3\log(\log(T))}{2T_{\boldsymbol{a}}(t)}} \\
&= -\Delta_{\boldsymbol{a}} + \frac{2\Delta_{\boldsymbol{a}}}{2} + K_{\boldsymbol{a}}\sqrt{\frac{\log(T)+3\log(\log(T))}{2T_{\boldsymbol{a}}(t)}}. \\
&= -\frac{\Delta_{\boldsymbol{a}}}{2} + K_{\boldsymbol{a}}\sqrt{\frac{\log(T)+3\log(\log(T))}{2T_{\boldsymbol{a}}(t)}}.
\end{aligned}
$$

Hence, $T_{\boldsymbol{a}}(t) < K_{\boldsymbol{a}}^2 \frac{2\log(T)+6\log(\log(T))}{\Delta_{\boldsymbol{a}}^2}$.
Therefore, $C \subseteq \left\{t \in [T] \setminus [|E|] : \boldsymbol{a}(t) = \boldsymbol{a}, T_{\boldsymbol{a}}(t) < K_{\boldsymbol{a}}^2 \frac{2\log(T)+6\log(\log(T))}{\Delta_{\boldsymbol{a}}^2}\right\}$, and

$$
\begin{aligned}
&\mathbb{E}\left[\sum_{t=|E|}^{T} \mathbb{1}\left\{\boldsymbol{a}(t) = \boldsymbol{a}, \forall e \in \boldsymbol{a}\setminus\boldsymbol{a}^*, |\hat{\rho}_e(t) - \rho_e| < \frac{\Delta_{\boldsymbol{a}}}{2K_{\boldsymbol{a}}}, \forall e \in \boldsymbol{a}^*\setminus\boldsymbol{a}, b_e(t) > \rho_e\right\}\right] \\
&= \mathbb{E}\left[|C|\right] \\
&\leqslant \mathbb{E}\left[\left|\left\{t \in [T] \setminus [|E|] : \boldsymbol{a}(t) = \boldsymbol{a}, T_{\boldsymbol{a}}(t) < K_{\boldsymbol{a}}^2 \frac{2\log(T)+6\log(\log(T))}{\Delta_{\boldsymbol{a}}^2}\right\}\right|\right] \\
&\leqslant K_{\boldsymbol{a}}^2 \frac{2\log(T)+6\log(\log(T))}{\Delta_{\boldsymbol{a}}^2}.
\end{aligned}
$$

Regret upper-bound Overall,

$$
\begin{aligned}
\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{\boldsymbol{a}(t) = \boldsymbol{a}\}\right] &\leqslant \frac{4K_{\boldsymbol{a}}^3}{\Delta_{\boldsymbol{a}}^2} + \mathcal{O}(\bar{K}_{\boldsymbol{a}} \log\log T) + K_{\boldsymbol{a}}^2 \frac{2\log(T)+6\log(\log(T))}{\Delta_{\boldsymbol{a}}^2} + |E| \\
&= \frac{2K_{\boldsymbol{a}}^2}{\Delta_{\boldsymbol{a}}^2}\log(T) + \mathcal{O}\left(\left(\bar{K}_{\boldsymbol{a}} + \frac{K_{\boldsymbol{a}}^2}{\Delta_{\boldsymbol{a}}^2}\right)\log\log T\right)
\end{aligned}
$$

and

$$
\begin{aligned}
R(T) &= \sum_{\boldsymbol{a}\in\mathcal{A}:\mu_{\boldsymbol{a}}\neq\mu^*} \Delta_{\boldsymbol{a}} \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{\boldsymbol{a}(t)=\boldsymbol{a}\}\right] \\
&\leqslant \sum_{\boldsymbol{a}\in\mathcal{A}:\mu_{\boldsymbol{a}}\neq\mu^*} \frac{2K_{\boldsymbol{a}}^2}{\Delta_{\boldsymbol{a}}} \log(T) + \mathcal{O}\left(\left(\bar{K}_{\boldsymbol{a}}\Delta_{\boldsymbol{a}} + \frac{K_{\boldsymbol{a}}^2}{\Delta_{\boldsymbol{a}}}\right)\log\log T\right) \\
&= \mathcal{O}\left(\frac{|\mathcal{A}|K_{max}^2}{\Delta_{\min}}\log T\right),
\end{aligned}
$$

which concludes the proof.

□

A.5 Proof of Lemma 2 (Upper-bound on the Number of Iterations of GRAB for which $\tilde{\boldsymbol{a}}(t)=\tilde{\boldsymbol{a}}\neq\boldsymbol{a}^*$)

Proof of Lemma 2. Let $\tilde{\boldsymbol{a}}\in\mathcal{P}_K^L\backslash\{\boldsymbol{a}^*\}$ and prove that $\mathbb{E}\left[\sum_{t=1}^{T}\mathbb{1}\{\tilde{\boldsymbol{a}}(t)=\tilde{\boldsymbol{a}}\}\right]=\mathcal{O}\left(\log\log T\right)$.

The proof requires notations related to the neighborhood of $\tilde{\boldsymbol{a}}$. Let $\mathcal{N}\overset{def}{=}\bigcup_{\boldsymbol{\pi}\in\mathcal{P}_K^K}\mathcal{N}_{\boldsymbol{\pi}}(\tilde{\boldsymbol{a}})$ be the set of all the potential neighbors of $\tilde{\boldsymbol{a}}$. By definition of the neighborhoods,

$$
\mathcal{N}=\left\{\tilde{\boldsymbol{a}}\circ(k,k') : k,k'\in[K]^2, k>k'\right\}\cup\left\{\tilde{\boldsymbol{a}}[k:=i] : k\in[K], i\in[L]\setminus\tilde{\boldsymbol{a}}([K])\right\},
$$

and its size is $N=K(2L-K-1)/2$. As $\tilde{\boldsymbol{a}}$ is sub-optimal, and due to Assumption 1, for any appropriate ranking of positions $\boldsymbol{\pi}\in\Pi_{\boldsymbol{\rho}}(\tilde{\boldsymbol{a}})$, there exists a recommendation $\boldsymbol{a}^+$ with a strictly better expected reward than $\tilde{\boldsymbol{a}}$ in the neighborhood $\mathcal{N}_{\boldsymbol{\pi}}(\tilde{\boldsymbol{a}})$. We denote

$$
\mathcal{N}^+\overset{def}{=}\bigcup_{\boldsymbol{\pi}\in\Pi_{\boldsymbol{\rho}}(\tilde{\boldsymbol{a}})}\left\{\boldsymbol{a}^+\in\mathcal{N}_{\boldsymbol{\pi}}(\tilde{\boldsymbol{a}}) : \mu_{\boldsymbol{a}^+}=\max_{\boldsymbol{a}\in\mathcal{N}_{\boldsymbol{\pi}}(\tilde{\boldsymbol{a}})}\mu_{\boldsymbol{a}}\right\}
$$

the set of such recommendations. We also chose $\epsilon<\min\{1/(2N),1/L\}$ and note

$$
\delta\overset{def}{=}\min_{\boldsymbol{\pi}\in\Pi_{\boldsymbol{\rho}}(\tilde{\boldsymbol{a}})}\min_{\boldsymbol{a}\in\mathcal{N}_{\boldsymbol{\pi}}(\tilde{\boldsymbol{a}})\cup\{\tilde{\boldsymbol{a}}\}\backslash\mathcal{N}^+}\left(\max_{\boldsymbol{a}'\in\mathcal{N}_{\boldsymbol{\pi}}(\tilde{\boldsymbol{a}})}\mu_{\boldsymbol{a}'}-\mu_{\boldsymbol{a}}\right).
$$

To bound $\mathbb{E}\left[\mathbb{1}\{\tilde{\boldsymbol{a}}(t)=\tilde{\boldsymbol{a}}\}\right]$, we use the decomposition $\{t\in[T]:\tilde{\boldsymbol{a}}(t)=\tilde{\boldsymbol{a}}\}\subseteq\bigcup_{\boldsymbol{a}^+\in\mathcal{N}^+}A_{\boldsymbol{a}^+}\cup$

B where for any permutation $\boldsymbol{a}^+ \in \mathcal{N}^+$,

$$A_{\boldsymbol{a}^+} = \{t : \tilde{\boldsymbol{a}}(t) = \tilde{\boldsymbol{a}}, T_{\boldsymbol{a}^+}(t) \geqslant \epsilon \tilde{T}_{\tilde{\boldsymbol{a}}}(t)\}$$

and

$$B = \{t : \tilde{\boldsymbol{a}}(t) = \tilde{\boldsymbol{a}}, \forall \boldsymbol{a}^+ \in \mathcal{A}+, T_{\boldsymbol{a}^+}(t) < \epsilon \tilde{T}_{\tilde{\boldsymbol{a}}}(t)\}.$$

Hence,

$$\mathbb{E}\left[\mathbb{1}\{\tilde{\boldsymbol{a}}(t) = \tilde{\boldsymbol{a}}\}\right] \leqslant \sum_{\boldsymbol{a}^+ \in \mathcal{A}+} \mathbb{E}\left[|A_{\boldsymbol{a}^+}|\right] + \mathbb{E}\left[|B|\right].$$

Bound on $\mathbb{E}\left[|A_{\boldsymbol{a}^+}|\right]$ Let $\boldsymbol{a}^+$ be a permutation in $\mathcal{N}^+$ and denote $\mathcal{K}^+$ the set of positions for which $\boldsymbol{a}^+$ and $\tilde{\boldsymbol{a}}$ disagree: $\mathcal{K}^+ = \left\{k \in [K] : a_k^+ \neq \tilde{a}_k\right\}$. The permutation $\boldsymbol{a}^+$ is in the neighborhood of $\tilde{\boldsymbol{a}}$, so either $\boldsymbol{a}^+ = \tilde{\boldsymbol{a}} \circ (k, k')$ or $\boldsymbol{a}^+ = \boldsymbol{a}[k := i]$, with k and k' in $[K]$, and i in $[L]$. Overall, $|\mathcal{K}^+| \leqslant 2$.

By the design of the algorithm and by definition of ϵ, we have that $\forall t \in A_{\boldsymbol{a}^+}$, $T_{\tilde{\boldsymbol{a}}}(t) \geqslant \tilde{T}_{\tilde{\boldsymbol{a}}}(t)/L > \epsilon \tilde{T}_{\tilde{\boldsymbol{a}}}(t)$. Moreover, at the considered iterations $\tilde{\boldsymbol{a}}$ is the leader, so

$$\begin{aligned}
A_{\boldsymbol{a}^+} &\subseteq \left\{t : \tilde{\boldsymbol{a}}(t) = \tilde{\boldsymbol{a}}, \tilde{T}_{\tilde{\boldsymbol{a}}}(t) < \frac{1}{\epsilon}\right\} \\
&\quad \cup \left\{t : \tilde{\boldsymbol{a}}(t) = \tilde{\boldsymbol{a}}, \min\{T_{\tilde{\boldsymbol{a}}}(t), T_{\boldsymbol{a}^+}(t)\} \geqslant \epsilon \tilde{T}_{\tilde{\boldsymbol{a}}}(t) \geqslant 1, \sum_{\ell} \hat{\rho}_{\tilde{a}_\ell, \ell}(t) \geqslant \sum_{\ell} \hat{\rho}_{a_\ell^+, \ell}(t)\right\} \\
&\subseteq \left\{t : \tilde{\boldsymbol{a}}(t) = \tilde{\boldsymbol{a}}, \tilde{T}_{\tilde{\boldsymbol{a}}}(t) < \frac{1}{\epsilon}\right\} \\
&\quad \cup \left\{t : \tilde{\boldsymbol{a}}(t) = \tilde{\boldsymbol{a}}, \min\{T_{\tilde{\boldsymbol{a}}}(t), T_{\boldsymbol{a}^+}(t)\} \geqslant \epsilon \tilde{T}_{\tilde{\boldsymbol{a}}}(t), \sum_{k \in \mathcal{K}^+} \hat{\rho}_{\tilde{a}_k, k}(t) \geqslant \sum_{k \in \mathcal{K}^+} \hat{\rho}_{a_k^+, k}(t)\right\} \\
&\subseteq \left\{t : \tilde{\boldsymbol{a}}(t) = \tilde{\boldsymbol{a}}, \tilde{T}_{\tilde{\boldsymbol{a}}}(t) < \frac{1}{\epsilon}\right\} \\
&\quad \cup \Big\{t : \tilde{\boldsymbol{a}}(t) = \tilde{\boldsymbol{a}}, \min\{T_{\tilde{\boldsymbol{a}}}(t), T_{\boldsymbol{a}^+}(t)\} \geqslant \epsilon \tilde{T}_{\tilde{\boldsymbol{a}}}(t), \exists k \in \mathcal{K}^+, \\
&\qquad\qquad |\hat{\rho}_{\tilde{a}_k, k}(t) - \rho_{\tilde{a}_k, k}| \geqslant \frac{\delta}{2|\mathcal{K}^+|} \text{ or } |\hat{\rho}_{a_k^+, k}(t) - \rho_{a_k^+, k}| \geqslant \frac{\delta}{2|\mathcal{K}^+|}\Big\} \\
&\subseteq \left\{t : \tilde{\boldsymbol{a}}(t) = \tilde{\boldsymbol{a}}, \tilde{T}_{\tilde{\boldsymbol{a}}}(t) < \frac{1}{\epsilon}\right\} \cup \bigcup_{k \in \mathcal{K}^+} \bigcup_{i \in \left\{\tilde{a}_k, a_k^+\right\}} \Lambda_{i,k},
\end{aligned}$$

with $\Lambda_{i,k} \stackrel{def}{=} \left\{t : \tilde{\boldsymbol{a}}(t) = \tilde{\boldsymbol{a}}, \min\{T_{\tilde{\boldsymbol{a}}}(t), T_{\boldsymbol{a}^+}(t)\} \geqslant \epsilon \tilde{T}_{\tilde{\boldsymbol{a}}}(t), |\hat{\rho}_{i,k}(t) - \rho_{i,k}| \geqslant \frac{\delta}{2|\mathcal{K}^+|}\right\}$.

Fix k in $\mathcal{K}^+$ and i in $\left\{\tilde{a}_k, a_k^+\right\}$. $\Lambda_{i,k} \subseteq \bigcup_{s \in \mathbb{N}} \Lambda_{i,k}(s)$, with $\Lambda_{i,k}(s) \stackrel{def}{=} \{t \in \Lambda_{i,k} : \tilde{T}_{\tilde{\boldsymbol{a}}}(t) = s\}$. $|\Lambda_{i,k}(s)| \leqslant 1$ as $\tilde{T}_{\tilde{\boldsymbol{a}}}(t)$ increases for each $t \in \Lambda_{i,k}$. Note that for each $s \in \mathbb{N}$ and

$n \in \Lambda_{i,k}(s)$, $T_{i,k}(n) \geqslant \min\{T_{\boldsymbol{a}}(n), T_{\boldsymbol{a}^+}(n)\} \geqslant \epsilon \tilde{T}_{\tilde{\boldsymbol{a}}}(n) = \epsilon s$. Then, by Lemma 6

$$\begin{aligned}
\mathbb{E}\left[|\Lambda_{i,k}|\right] &= \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{t \in \Lambda_{i,k}\}\right] \\
&= \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\left\{t \in \Lambda_{i,k}, |\hat{\rho}_{i,k}(t) - \rho_{i,k}| > \frac{\delta}{2|\mathcal{K}^+|}\right\}\right] \\
&\leqslant \frac{4|\mathcal{K}^+|^2}{\epsilon\delta^2}
\end{aligned}$$

Hence, $\mathbb{E}\left[|A_{\boldsymbol{a}^+}|\right] \leqslant \frac{1}{\epsilon} + \sum_{k\in\mathcal{K}^+} \sum_{i\in\left\{\tilde{a}_k, a_k^+\right\}} \mathbb{E}\left[|\Lambda_{i,k}|\right] \leqslant \frac{1}{\epsilon} + \frac{8|\mathcal{K}^+|^3}{\epsilon\delta^2}$.

Bound on $\mathbb{E}\left[|B|\right]$ We first split B in two parts: $B = B^{t_0} \cup B_{t_0}^T$, where $B^{t_0} \overset{def}{=} \{t \in B : \tilde{T}_{\tilde{\boldsymbol{a}}}(t) \leqslant t_0\}$, $B_{t_0}^T \overset{def}{=} \{t \in B : \tilde{T}_{\tilde{\boldsymbol{a}}}(t) > t_0\}$, and t_0 is chosen as small as possible to satisfy three constraints required in the rest of the proof.

Namely, $t_0 = \max\left\{\frac{1}{\epsilon}, (1+N)(1-\frac{1}{L}-\epsilon N)^{-1}, \inf\left\{t : 2\sqrt{\frac{\log(t+1)+3\log(\log(t+1))}{2\epsilon t}} < \frac{\delta}{8}\right\}\right\}$. Note that t_0 only depends on K, L and δ, and that $(1-\frac{1}{L}-\epsilon N) > 0$ (assuming $L \geqslant 2$) as $\epsilon < 1/(2N)$.

We also define

— $D \overset{def}{=} \bigcup_{(\boldsymbol{a},k)\in(\mathcal{N}\cup\{\tilde{\boldsymbol{a}}\}\setminus\mathcal{N}^+)\times[K]} D_{\boldsymbol{a},k}$,
where $D_{\boldsymbol{a},k} \overset{def}{=} \left\{t \in [T] : \tilde{\boldsymbol{a}}(t) = \tilde{\boldsymbol{a}}, \boldsymbol{a}(t) = \boldsymbol{a}, |\hat{\rho}_{a_k,k}(t) - \rho_{a_k,k}| \geqslant \frac{\delta}{8}\right\}$,
— $E \overset{def}{=} \bigcup_{(\boldsymbol{a}^+,k)\in\mathcal{N}^+\times[K]} E_{\boldsymbol{a}^+,k}$, where $E_{\boldsymbol{a}^+,k} \overset{def}{=} \{t \in [T] : \tilde{\boldsymbol{a}}(t) = \tilde{\boldsymbol{a}}, b_{a_k^+,k}(t) \leqslant \rho_{a_k^+,k}\}$,
— and $F \overset{def}{=} \{t \in [T] : \tilde{\boldsymbol{a}}(t) = \tilde{\boldsymbol{a}}, \tilde{\boldsymbol{\pi}}(t) \notin \Pi_{\boldsymbol{\rho}}(\tilde{\boldsymbol{a}})\}$.

Let $t \in B_{t_0}^T$. By construction, GRAB forces itself to select $\left\lceil \frac{\tilde{T}_{\tilde{\boldsymbol{a}}}(t)}{L} \right\rceil$ times the leader $\tilde{\boldsymbol{a}}$ between iterations 1 and $t-1$. So,

$$\tilde{T}_{\tilde{\boldsymbol{a}}}(t) = \left\lceil \frac{\tilde{T}_{\tilde{\boldsymbol{a}}}(t)}{L} \right\rceil + \sum_{\boldsymbol{a}\in\mathcal{N}\cup\{\tilde{\boldsymbol{a}}\}} T_{\boldsymbol{a}}^{\tilde{\boldsymbol{a}}}(t)$$

where $T_{\boldsymbol{a}}^{\tilde{\boldsymbol{a}}}(t) = \sum_{s=1}^{t-1} \mathbb{1}\left\{\tilde{\boldsymbol{a}}(s) = \tilde{\boldsymbol{a}}, \boldsymbol{a}(s) = \boldsymbol{a}, \tilde{T}_{\tilde{\boldsymbol{a}}}(s)/L \notin \mathbb{N}\right\}$ is the number of times arm $\boldsymbol{a} \in \mathcal{N}\cup\{\tilde{\boldsymbol{a}}\}$ has been played **normally** (i.e not forced) while $\tilde{\boldsymbol{a}}$ was leader, up to time $t-1$. Let prove by contradiction that there is at least one recommendation $\boldsymbol{a}$ that has been selected **normally** more than $\epsilon\tilde{T}_{\tilde{\boldsymbol{a}}}(t) + 1$ times, namely $T_{\boldsymbol{a}}^{\tilde{\boldsymbol{a}}}(t) \geqslant \epsilon\tilde{T}_{\tilde{\boldsymbol{a}}}(t) + 1$.

Assume that for each recommendation $\boldsymbol{a}$ in $\mathcal{N} \cup \{\tilde{\boldsymbol{a}}\}$, $T_{\boldsymbol{a}}^{\tilde{\boldsymbol{a}}}(t) < \epsilon \tilde{T}_{\tilde{\boldsymbol{a}}}(t) + 1$. Then

$$\begin{aligned}\tilde{T}_{\tilde{\boldsymbol{a}}}(t) &= \left\lceil \frac{\tilde{T}_{\tilde{\boldsymbol{a}}}(t)}{L} \right\rceil + \sum_{\boldsymbol{a} \in \mathcal{N} \cup \{\tilde{\boldsymbol{a}}\}} T_{\boldsymbol{a}}^{\tilde{\boldsymbol{a}}}(t) \\ &< 1 + \frac{\tilde{T}_{\tilde{\boldsymbol{a}}}(t)}{L} + N(\epsilon \tilde{T}_{\tilde{\boldsymbol{a}}}(t) + 1).\end{aligned}$$

Therefore $\tilde{T}_{\tilde{\boldsymbol{a}}}(t)(1 - \frac{1}{L} - N\epsilon) < 1 + N$, which contradicts $t \in B_{t_0}^T$.

So, there exists a recommendation $\boldsymbol{a}$ such that $T_{\boldsymbol{a}}^{\tilde{\boldsymbol{a}}}(t) \geqslant \epsilon \tilde{T}_{\tilde{\boldsymbol{a}}}(t) + 1$. Let denote s' the first iteration such that $T_{\boldsymbol{a}}^{\tilde{\boldsymbol{a}}}(s') \geqslant \epsilon \tilde{T}_{\tilde{\boldsymbol{a}}}(t) + 1$. At this iteration, $T_{\boldsymbol{a}}^{\tilde{\boldsymbol{a}}}(s') = T_{\boldsymbol{a}}^{\tilde{\boldsymbol{a}}}(s' - 1) + 1$, meaning that $\tilde{\boldsymbol{a}}(s' - 1) = \tilde{\boldsymbol{a}}$, $\boldsymbol{a}(s' - 1) = \boldsymbol{a}$, $\tilde{T}_{\tilde{\boldsymbol{a}}}(s' - 1)/L \notin \mathbb{N}$, and $T_{\boldsymbol{a}}^{\tilde{\boldsymbol{a}}}(s' - 1) \geqslant \epsilon \tilde{T}_{\tilde{\boldsymbol{a}}}(t)$. Therefore, the set $\{s \in [t] : \tilde{\boldsymbol{a}}(s) = \tilde{\boldsymbol{a}}, T_{\boldsymbol{a}(s)}^{\tilde{\boldsymbol{a}}}(s) \geqslant \epsilon \tilde{T}_{\tilde{\boldsymbol{a}}}(t), \tilde{T}_{\tilde{\boldsymbol{a}}}(s)/L \notin \mathbb{N}\}$ is non-empty. We define $\psi(t)$ as the minimum on this set

$$\psi(t) \stackrel{def}{=} \min \left\{ s \in [t] : \tilde{\boldsymbol{a}}(s) = \tilde{\boldsymbol{a}}, T_{\boldsymbol{a}(s)}^{\tilde{\boldsymbol{a}}}(s) \geqslant \epsilon \tilde{T}_{\tilde{\boldsymbol{a}}}(t), \tilde{T}_{\tilde{\boldsymbol{a}}}(s)/L \notin \mathbb{N} \right\}.$$

We note $\boldsymbol{a}$ the recommendation $\boldsymbol{a}(\psi(t))$ at iteration $\psi(t)$. We have $\boldsymbol{a} \notin \mathcal{N}^+$ since for any recommendation $\boldsymbol{a}^+ \in \mathcal{N}^+$, $T_{\boldsymbol{a}^+}^{\tilde{\boldsymbol{a}}}(\psi(t)) \leqslant T_{\boldsymbol{a}^+}^{\tilde{\boldsymbol{a}}}(t) \leqslant T_{\boldsymbol{a}^+}(t) < \epsilon \tilde{T}_{\tilde{\boldsymbol{a}}}(t)$. Let $\boldsymbol{a}^+$ be one of the best recommendations in $\mathcal{N}_{\tilde{\boldsymbol{\pi}}(\psi(t))}(\tilde{\boldsymbol{a}}) \cup \{\tilde{\boldsymbol{a}}\}$, meaning $\mu_{\boldsymbol{a}^+} = \max_{\boldsymbol{a}' \in \mathcal{N}_{\tilde{\boldsymbol{\pi}}(\psi(t))}(\tilde{\boldsymbol{a}}) \cup \{\tilde{\boldsymbol{a}}\}} \mu_{\boldsymbol{a}'}$, and let $\mathcal{K}$ denote the set of positions for which $\boldsymbol{a}$ and $\boldsymbol{a}^+$ disagree. As both recommendations are in $\mathcal{N}_{\tilde{\boldsymbol{\pi}}(\psi(t))}(\tilde{\boldsymbol{a}}) \cup \{\tilde{\boldsymbol{a}}\}$, $|\mathcal{K}| \leqslant 4$.

Let prove by contradiction that $\psi(t) \in D \cup E \cup F$. Assume that $\psi(t) \notin D \cup E \cup F$.

Since $\psi(t) \notin F$, $\tilde{\boldsymbol{\pi}}(\psi(t))$ belongs to $\Pi_{\boldsymbol{\rho}}(\tilde{\boldsymbol{a}})$ and hence $\boldsymbol{a}^+$ is in $\mathcal{N}^+$ and $\sum_k \rho_{a_k^+,k} - \sum_k \rho_{a_k,k} = \mu_{\boldsymbol{a}^+} - \mu_{\boldsymbol{a}} \geqslant \delta$.

Moreover, since $\psi(t) \notin D \cup E$, for each position $k \in [K]$, $|\hat{\rho}_{a_k,k}(\psi(t)) - \rho_{a_k,k}| < \frac{\delta}{8}$, and $b_{a_k^+,k}(\psi(t)) > \rho_{a_k^+,k}$.

Finally, $T_{\boldsymbol{a}}(\psi(t)) \geqslant T_{\boldsymbol{a}}^{\tilde{\boldsymbol{a}}}(\psi(t)) \geqslant \epsilon \tilde{T}_{\tilde{\boldsymbol{a}}}(t) \geqslant 1$, and therefore $b_{a_k,k}(\psi(t))$ and $\hat{\rho}_{a_k,k}(\psi(t))$ are properly defined for any position $k \in [K]$.

Then, by Pinsker's inequality and the fact that $\psi(t) \leqslant t$, $\tilde{T}_{\tilde{\boldsymbol{a}}}(s)$ is non-decreasing in s,

and $T_{\boldsymbol{a}}(\psi(t)) \geqslant \epsilon \tilde{T}_{\tilde{\boldsymbol{a}}}(t)$,

$$\begin{aligned}
\sum_k b_{a_k,k}(\psi(t)) - \sum_k b_{a_k^+,k}(\psi(t)) &= \sum_{k\in\mathcal{K}} b_{a_k,k}(\psi(t)) - b_{a_k^+,k}(\psi(t)) \\
&\leqslant \sum_{k\in\mathcal{K}} \hat{\rho}_{a_k,k}(\psi(t)) + \sqrt{\frac{\log(\tilde{T}_{\tilde{\boldsymbol{a}}}(\psi(t))+1) + 3\log(\log(\tilde{T}_{\tilde{\boldsymbol{a}}}(\psi(t))+1))}{2T_{\boldsymbol{a}}(\psi(t))}} \\
&\qquad - b_{a_k^+,k}(\psi(t)) \\
&< \sum_{k\in\mathcal{K}} \rho_{a_k,k} + \frac{\delta}{8} + \sqrt{\frac{\log(\tilde{T}_{\tilde{\boldsymbol{a}}}(t)+1) + 3\log(\log(\tilde{T}_{\tilde{\boldsymbol{a}}}(t)+1))}{2\epsilon\tilde{T}_{\tilde{\boldsymbol{a}}}(t)}} - \rho_{a_k^+,k} \\
&\leqslant \sum_{k\in\mathcal{K}} \rho_{a_k,k} + \frac{\delta}{8} + \frac{\delta}{8} - \rho_{a_k^+,k} \\
&\leqslant \sum_k \rho_{a_k,k} - \sum_k \rho_{a_k^+,k} + |\mathcal{K}| \cdot 2\frac{\delta}{8} \\
&\leqslant -\delta + 8\frac{\delta}{8} \\
&= 0,
\end{aligned}$$

which contradicts the fact that $\boldsymbol{a}$ is played at iteration $\psi(t)$. So $\psi(t) \in D \cup E \cup F$.

Overall, for any $t \in B_{t_0}^T$, $\psi(t) \in D \cup E \cup F$. So, $B_{t_0}^T \subseteq \bigcup_{n\in D\cup E\cup F} B_{t_0}^T \cap \{t \in [T] : \psi(t) = n\}$. Let n be in $D \cup E \cup F$. For any t in $B_{t_0}^T \cap \{t \in [T] : \psi(t) = n\}$, $T_{\boldsymbol{a}(n)}^{\tilde{\boldsymbol{a}}}(n) = \lceil \epsilon \tilde{T}_{\tilde{\boldsymbol{a}}}(t) \rceil$ and $\tilde{T}_{\tilde{\boldsymbol{a}}}(t+1) = \tilde{T}_{\tilde{\boldsymbol{a}}}(t) + 1$. So $|B_{t_0}^T \cap \{t \in [T] : \psi(t) = n\}| < 1/\epsilon + 1$. Overall,

$$\mathbb{E}\left[|B|\right] \leqslant t_0 + \mathbb{E}\left[|B_{t_0}^T|\right] \leqslant t_0 + (1/\epsilon + 1)\left(\mathbb{E}\left[|D|\right] + \mathbb{E}\left[|E|\right] + \mathbb{E}\left[|F|\right]\right).$$

It remains to upper-bound $\mathbb{E}\left[|D|\right]$, $\mathbb{E}\left[|E|\right]$, and $\mathbb{E}\left[|F|\right]$ to conclude the proof.

Bound on $\mathbb{E}\left[|D|\right]$ The upper-bound on $\mathbb{E}\left[|D|\right]$ is obtained with the same strategy as the last step in the proof of the upper-bound on $\mathbb{E}\left[|A_{\boldsymbol{a}^+}|\right]$. Let $\boldsymbol{a}$ be a recommendation in $\mathcal{N} \cup \{\tilde{\boldsymbol{a}}\} \setminus \mathcal{N}^+$, and $k \in [K]$ be a position. $D_{\boldsymbol{a},k} \subseteq \bigcup_{s\in\mathbb{N}} \Lambda_{\boldsymbol{a},k}(s)$, where $\Lambda_{\boldsymbol{a},k}(s) \stackrel{def}{=} \{t \in D_{\boldsymbol{a},k} : T_{\boldsymbol{a}}(t) = s\}$. $|\Lambda_{\boldsymbol{a},k}(s)| \leqslant 1$ as $T_{\boldsymbol{a}}(t)$ increases for each $t \in D_{\boldsymbol{a},k}$. Note that for each $s \in \mathbb{N}$ and $n \in \Lambda_{\boldsymbol{a},k}(s)$, $T_{a_k,k}(n) \geqslant T_{\boldsymbol{a}}(n) = s$. Then, by Lemma 6

$$\begin{aligned}\mathbb{E}\left[|D_{\boldsymbol{a},k}|\right] &\leq \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{t \in D_{\boldsymbol{a},k}\}\right] \\ &= \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\left\{t \in D_{\boldsymbol{a},k}, |\hat{\rho}_{a_k,k}(t) - \rho_{a_k,k}| \geqslant \frac{\delta}{8}\right\}\right] \\ &\leqslant \frac{64}{\delta^2}\end{aligned}$$

Hence, $\mathbb{E}\left[|D|\right] \leq \sum_{(\boldsymbol{a},k)\in(\mathcal{N}\cup\{\tilde{\boldsymbol{a}}\}\setminus\mathcal{N}^+)\times[K]} \mathbb{E}\left[|D_{\boldsymbol{a},k}|\right] \leqslant \frac{64(N+1)K}{\delta^2}$.

Bound on $\mathbb{E}\left[|E|\right]$ By Theorem 10 of [21], $\mathbb{E}\left[|E_{\boldsymbol{a}^+,k}|\right] = O(\log(\log(T)))$, so $\mathbb{E}\left[|E|\right] \leqslant \sum_{(\boldsymbol{a}^+,k)\in\mathcal{N}^+\times[K]} \mathbb{E}\left[|E_{\boldsymbol{a}^+,k}|\right] = O(|\mathcal{N}^+|K\log(\log(T)))$.

Bound on $\mathbb{E}\left[|F|\right]$ By Lemma 3, $\mathbb{E}\left[|F|\right] = \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\left\{\tilde{\boldsymbol{a}}(t) = \tilde{\boldsymbol{a}}, \tilde{\boldsymbol{\pi}}(t) \notin \Pi_{\boldsymbol{\rho}}\left(\tilde{\boldsymbol{a}}\right)\right\}\right] = \mathcal{O}\left(1\right)$.

Overall $\mathbb{E}\left[\mathbb{1}\{\tilde{\boldsymbol{a}}(t) = \tilde{\boldsymbol{a}}\}\right] \leqslant \frac{|\mathcal{K}^+|}{\epsilon} + \frac{8|\mathcal{K}^+|^3|\mathcal{N}^+|}{\epsilon\delta^2} + t_0 + \left(\frac{1}{\epsilon}+1\right)\frac{64(N+1)K}{\delta^2} + \mathcal{O}\left(\frac{|\mathcal{N}^+|K}{\epsilon}\log\log T\right) + \mathcal{O}(1) = \mathcal{O}\left(\frac{|\mathcal{N}^+|K}{\epsilon}\log\log T\right)$, which concludes the proof. □

A.6 Proof of Lemma 3 (Upper-bound on the Number of Iterations of GRAB for which $\tilde{\boldsymbol{\pi}}(t) \notin \Pi_{\boldsymbol{\rho}}(\tilde{\boldsymbol{a}})$)

Proof of Theorem 3. Let $\tilde{\boldsymbol{a}}$ be a K-permutation of L items. If $\Pi_{\boldsymbol{\rho}}\left(\tilde{\boldsymbol{a}}\right)$ contains all the permutations of K elements, the set $\{t : \tilde{\boldsymbol{a}}(t) = \tilde{\boldsymbol{a}}, \tilde{\boldsymbol{\pi}}(t) \notin \Pi_{\boldsymbol{\rho}}\left(\tilde{\boldsymbol{a}}\right)\}$ is empty.

Otherwise, let denote δ the smallest non-zero gap between the probability of click at position k and the probability of click at position $k' \neq k$: $\delta \stackrel{def}{=} \min\left\{\rho_{\tilde{a}_k,k} - \rho_{\tilde{a}_{k'},k'} : (k,k') \in [K]^2, \rho_{\tilde{a}_k,k} - \rho_{\tilde{a}_{k'},k'} > 0\right\}$. The gap δ is the minimum on a finite set, so $\delta > 0$.

By definition of $\tilde{\pi}(t)$, $\hat{\rho}_{\tilde{a}_{\tilde{\pi}_1(t)}(t),\tilde{\pi}_1(t)}(t) \geqslant \hat{\rho}_{\tilde{a}_{\tilde{\pi}_2(t)}(t),\tilde{\pi}_2(t)}(t) \geqslant \cdots \geqslant \hat{\rho}_{\tilde{a}_{\tilde{\pi}_K(t)}(t),\tilde{\pi}_K(t)}(t)$, so,

$$\begin{aligned}\{t : \tilde{\boldsymbol{a}}(t) = \tilde{\boldsymbol{a}}, \tilde{\boldsymbol{\pi}}(t) \notin \Pi_{\boldsymbol{\rho}}\left(\tilde{\boldsymbol{a}}\right)\} &= \bigcup_{\tilde{\boldsymbol{\pi}}\in\mathcal{P}_K^K} \bigcup_{k\in[K-1]} \left\{t : \tilde{\boldsymbol{a}}(t) = \tilde{\boldsymbol{a}}, \tilde{\boldsymbol{\pi}}(t) = \tilde{\boldsymbol{\pi}}, \rho_{\tilde{a}_{\tilde{\pi}_k},\tilde{\pi}_k} < \rho_{\tilde{a}_{\tilde{\pi}_{k+1}},\tilde{\pi}_{k+1}}\right\} \\ &\subseteq \bigcup_{\tilde{\boldsymbol{\pi}}\in\mathcal{P}_K^K} \bigcup_{k\in[K-1]} \left\{t : \tilde{\boldsymbol{a}}(t) = \tilde{\boldsymbol{a}}, \tilde{\boldsymbol{\pi}}(t) = \tilde{\boldsymbol{\pi}}, \begin{array}{c}|\hat{\rho}_{\tilde{a}_{\tilde{\pi}_k},\tilde{\pi}_k}(t) - \rho_{\tilde{a}_{\tilde{\pi}_k},\tilde{\pi}_k}| > \frac{\delta}{2} \\ \text{or } |\hat{\rho}_{\tilde{a}_{\tilde{\pi}_{k+1}},\tilde{\pi}_{k+1}}(t) - \rho_{\tilde{a}_{\tilde{\pi}_{k+1}},\tilde{\pi}_{k+1}}| > \frac{\delta}{2}\end{array}\right\} \\ &= \bigcup_{\tilde{\boldsymbol{\pi}}\in\mathcal{P}_K^K} \bigcup_{k\in[K]} \Lambda_{\tilde{\boldsymbol{\pi}},k},\end{aligned}$$

with $\Lambda_{\tilde{\boldsymbol{\pi}},k} \overset{def}{=} \left\{t : \tilde{\boldsymbol{a}}(t) = \tilde{\boldsymbol{a}}, \tilde{\boldsymbol{\pi}}(t) = \tilde{\boldsymbol{\pi}}, |\hat{\rho}_{\tilde{a}_{\tilde{\pi}_k},\tilde{\pi}_k}(t) - \rho_{\tilde{a}_{\tilde{\pi}_k},\tilde{\pi}_k}| > \frac{\delta}{2}\right\}$, for any ranking of positions $\tilde{\boldsymbol{\pi}} \in \mathcal{P}_K^L$ and any rank $k \in [K]$.

Let $\tilde{\boldsymbol{\pi}} \in \mathcal{P}_K^L$ be a ranking of positions, and $k \in [K]$ be a rank. $\Lambda_{\tilde{\boldsymbol{\pi}},k} \subseteq \bigcup_{s\in\mathbb{N}} \Lambda_{\tilde{\boldsymbol{\pi}},k}(s)$, with $\Lambda_{\tilde{\boldsymbol{\pi}},k}(s) \overset{def}{=} \{t \in \Lambda_{\tilde{\boldsymbol{\pi}},k} : \tilde{T}_{\tilde{\boldsymbol{a}}}(t) = s\}$. $|\Lambda_{\tilde{\boldsymbol{\pi}},k}(s)| \leqslant 1$ as $\tilde{T}_{\tilde{\boldsymbol{a}}}(t)$ increases for each $t \in \Lambda_{\tilde{\boldsymbol{\pi}},k}$. Note that for each $s \in \mathbb{N}$ and $n \in \Lambda_{\tilde{\boldsymbol{\pi}},k}(s)$, $T_{\tilde{a}_{\tilde{\pi}_k},\tilde{\pi}_k}(n) \geqslant T_{\tilde{\boldsymbol{a}}}(n) \geqslant \tilde{T}_{\tilde{\boldsymbol{a}}}(n)/L = s/L$. Then, by Lemma 6

$$\begin{aligned}
\mathbb{E}\left[|\Lambda_{\tilde{\boldsymbol{\pi}},k}|\right] &= \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{t \in \Lambda_{\tilde{\boldsymbol{\pi}},k}\}\right] \\
&= \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\left\{t \in \Lambda_{\tilde{\boldsymbol{\pi}},k}, |\hat{\rho}_{\tilde{a}_{\tilde{\pi}_k},\tilde{\pi}_k}(t) - \rho_{\tilde{a}_{\tilde{\pi}_k},\tilde{\pi}_k}| > \frac{\delta}{2}\right\}\right] \\
&\leqslant \frac{4L}{\delta^2}
\end{aligned}$$

Hence,

$$\begin{aligned}
\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{\tilde{\boldsymbol{a}}(t) = \tilde{\boldsymbol{a}}, \tilde{\boldsymbol{\pi}}(t) \notin \Pi_{\boldsymbol{\rho}}(\tilde{\boldsymbol{a}})\}\right] &\leqslant \sum_{\tilde{\boldsymbol{\pi}}\in\mathcal{P}_K^K} \sum_{k\in[K]} \mathbb{E}\left[\Lambda_{\tilde{\boldsymbol{\pi}},k}\right] \\
&\leqslant \frac{4LKK!}{\delta^2} \\
&= \mathcal{O}\left(LKK!\right),
\end{aligned}$$

which concludes the proof.

□

A.7 S-GRAB: OSUB on a Static Graph

The algorithm S-GRAB, depicted in Algorithm 12, is similar to GRAB except that it explores a static graph $G = (E, V)$ defined by

$$\begin{aligned}
&V \overset{def}{=} \mathcal{P}_K^L, \\
&E \overset{def}{=} \left\{(\boldsymbol{a}, \boldsymbol{a} \circ (k, k')) : k, k' \in [K]^2, k > k'\right\} \cup \left\{(\boldsymbol{a}, \boldsymbol{a}[k := i]) : k \in [K], i \in [L] \setminus \boldsymbol{a}([K])\right\}.
\end{aligned}$$

This graph is chosen to ensure that with PBM setting any sub-optimal recommendation has a strictly better recommendation in its neighborhood given G. This graph is fixed and does not require the knowledge of a mapping $\mathcal{P}$, but its degree is also about K

Algorithm 12 S-GRAB: Static Graph for unimodal RAnking Bandit

Require: number of items L, number of positions K

$\gamma \leftarrow K(2L-K-1)/2$

for $t = 1, 2, \dots$ **do**

$\tilde{\boldsymbol{a}}(t) \leftarrow \underset{\boldsymbol{a} \in \mathcal{P}_K^L}{\operatorname{argmax}} \sum_{k=1}^{K} \hat{\rho}_{a_k,k}(t)$

$$\text{recommend } \boldsymbol{a}(t) = \begin{cases} \tilde{\boldsymbol{a}}(t) & \text{, if } \frac{\tilde{T}_{\tilde{\boldsymbol{a}}(t)}(t)}{\gamma+1} \in \mathbb{N}, \\ \underset{\boldsymbol{a} \in \{\tilde{\boldsymbol{a}}(t)\} \cup \mathcal{N}_G(\tilde{\boldsymbol{a}}(t))}{\operatorname{argmax}} \sum_{k=1}^{K} b_{a_k,k}(t) & \text{, otherwise} \end{cases}$$

where $\mathcal{N}_G(\boldsymbol{a}) = \{\boldsymbol{a} \circ (k,k') : k,k' \in [K]^2, k > k'\} \cup \{\boldsymbol{a}[k := i] : k \in [K], i \in [L] \setminus \boldsymbol{a}([K])\}$

observe the clicks vector $\boldsymbol{c}(t)$

end for

times larger than the degree of the graphs handled by GRAB.

As for GRAB, any recommendation in the neighborhood of the leader given G differs with the leader at, at most two positions. Therefore a proof similar to the one of Theorem 1 ensures that S-GRAB's regret is upper-bounded by $\mathcal{O}\left(LK/\Delta_{min} \log T\right)$. This regret upper-bound is higher than GRAB's one by a factor K due to the larger size of the considered neighborhoods. However, this regret remains smaller than KL-CombUCB's one by a factor K thanks to the bounded number of differences between the leader and the arm played.

Appendix B

UniRank

B.1 Organisation of the Appendix

The appendix is organized as follows. After listing most of the notations used in the paper in Appendix B.2, we prove Lemma 4 in Appendix B.3. Then we prove some technical lemmas in Appendix B.4, which are required by the proof of Theorem 3 in Appendix B.5. Finally, we discuss the regret upper-bound of UniRank for some specific settings in Appendix B.6.

B.2 Notations

Symbol	Meaning
T	Time horizon
t	iteration
L	number of items
i	index of an item
K	number of positions in a recommendation
k	index of a position
$[n]$	set of integers $\{1, \dots, n\}$
$\mathcal{P}_K^L$	set of permutations of K distinct items among L
$\boldsymbol{\theta}$	vectors of probabilities of click
θ_i	probability of click on item i
$\boldsymbol{\kappa}$	vectors of probabilities of view
κ_k	probability of view at position k
$\mathcal{A}$	set of bandit arms
$\boldsymbol{a}$	an arm in $\mathcal{A}$
$\boldsymbol{a}(t)$	the arm chosen at iteration t
a_k	item displayed at position k in the recommendation $\boldsymbol{a}$

Continued on next page

Symbol	Meaning
$\boldsymbol{a}^*$	BEST ARM
ρ	FUNCTION FROM $\mathcal{P}_K^L \times [K]$ TO $[0,1]$ GIVING THE PROBABILITY OF CLICK
$\rho(\boldsymbol{a}, k)$	PROBABILITY OF CLICK ON THE ITEM DISPLAYED AT POSITION k WHEN RECOMMENDING $\boldsymbol{a}$
$\boldsymbol{c}(t)$	CLICKS VECTOR AT ITERATION t
$c_i(t)$	CLICKS ON ITEM I AT ITERATION t
$r(t)$	REWARD COLLECTED AT ITERATION t, $r(t) = \sum_{i=1}^{L} c_i(t)$
$\mu_{\boldsymbol{a}}$	EXPECTATION OF $r(t)$ WHILE RECOMMENDING $\boldsymbol{a}$, $\mu_{\boldsymbol{a}} = \mathbb{E}[r(t) \mid \boldsymbol{a}(t) = \boldsymbol{a}]$
μ^*	HIGHEST EXPECTED REWARD, $\mu^* = \max_{\boldsymbol{a} \in \mathcal{P}_K^L} \mu_{\boldsymbol{a}}$
Δ	GENERIC REWARD GAP BETWEEN ONE OF THE SUB-OPTIMAL ARMS AND ONE OF THE BEST ARMS
Δ_c	REWARD GAP WHILE EXCHANGING ITEMS $\min(c-1, K)$ AND c IN THE OPTIMAL RECOMMENDATION,
$\tilde{\delta}_{i,j}$	SMALLEST PROBABILITY FOR $c_i(t)$ TO BE DIFFERENT FROM $c_j(t)$ WHILE BOTH ITEMS ARE IN THE SAME SUBSET OF THE CHOSEN PARTITION $\boldsymbol{P}(t)$
$\tilde{\delta}_k^*$	SMALLEST PROBABILITY FOR $c_{\min(k-1,K)}(t)$ TO BE DIFFERENT FROM $c_k(t)$, WHILE BOTH ITEMS ARE IN THE SAME SUBSET OF THE CHOSEN PARTITION $\boldsymbol{P}(t)$ AND $\boldsymbol{P}(t)$ IS IN THE NEIGHBORHOOD OF THE OPTIMAL PARTITION
$\tilde{\Delta}_{i,j}$	SMALLEST (RESPECTIVELY HIGHEST) EXPECTED DIFFERENCE OF CLICK BETWEEN ITEMS i AND j IF $i \succ j$ (RESP. $j \succ i$) WHILE BOTH ITEMS ARE IN THE SAME SUBSET OF THE CHOSEN PARTITION $\boldsymbol{P}(t)$
$R(T)$	CUMULATIVE (PSEUDO-)REGRET, $R(T) = T\mu^* - \mathbb{E}\left[\sum_{t=1}^{T} \mu_{\boldsymbol{a}(t)}\right]$
$\succ$	STRICT WEAK ORDER
$(i,j) \circ \boldsymbol{a}$	PERMUTATION SWAPPING ITEMS I AND J IN RECOMMENDATION $\boldsymbol{a}$
$\boldsymbol{P}$	ORDERED PARTITION OF ITEMS REPRESENTING A SUBSET OF RECOMMENDATIONS, $\boldsymbol{P} = (P_1, \ldots, P_d)$
P_c	c^{th} PART OF $\boldsymbol{P}$ SUCH AS $\bigcup_{c=1}^{d} P_c = [L]$, AND $P_c \cap P_{c'}$ IS EMPTY WHEN $c \neq c'$
$\mathcal{A}(\boldsymbol{P})$	SET OF RECOMMENDATIONS $\boldsymbol{a}$ AGREEING WITH $\boldsymbol{P}$
$\tilde{\boldsymbol{P}}(t)$	BEST PARTITION AT ITERATION t GIVEN THE PREVIOUS CHOICES AND FEEDBACKS (CALLED LEADER)
$\boldsymbol{P}^*$	PARTITION SUCH THAT ANY PERMUTATION $\boldsymbol{a}$ IN $\mathcal{A}(\boldsymbol{P}^*)$ IS COMPATIBLE WITH THE STRICT WEAK ORDER ON ITEMS.
$\mathcal{G}$	GRAPH CARRYING A PARTIAL ORDER ON THE PARTITIONS OF ITEMS

CONTINUED ON NEXT PAGE

Symbol	Meaning
$t_{i,j}(t)$	NUMBER OF ITERATIONS AT WHICH ITEMS i AND j HAVE BEEN GATHERED IN THE SAME SUBSET OF ITEMS $P_c(s)$, $t_{i,j}(t) \overset{def}{=} \sum_{s=1}^{t-1} \mathbb{1}\left\{\exists c, (i,j) \in P_c(s)^2\right\}$
$T_{i,j}(t)$	NUMBER OF ITERATIONS AT WHICH ITEMS i AND j HAVE BEEN GATHERED IN THE SAME SUBSET OF ITEMS $P_c(s)$ AND LEAD TO A DIFFERENT CLICK VALUE, $T_{i,j}(t) = \sum_{s=1}^{t-1} \mathbb{1}\left\{\exists c, (i,j) \in P_c(s)^2\right\} \mathbb{1}\{c_i(s) \neq c_j(s)\}$
$\tilde{t}_{\tilde{\boldsymbol{P}}}(t)$	NUMBER OF TIME A PERMUTATION $\tilde{\boldsymbol{P}}$ AS BEEN THE LEADER, $\tilde{t}_{\tilde{\boldsymbol{P}}}(t) \overset{def}{=} \sum_{s=1}^{t-1} \mathbb{1}\left\{\tilde{\boldsymbol{P}}(s) = \tilde{\boldsymbol{P}}\right\}$
$\tilde{\delta}_{i,j}(\boldsymbol{a})$	PROBABILITY OF DIFFERENCE,$\tilde{\delta}_{i,j}(\boldsymbol{a}) = \mathbb{P}_{\boldsymbol{a}' \sim \mathcal{U}(\{\boldsymbol{a},(i,j)\circ\boldsymbol{a}\})}\left[c_i \neq c_j\right]$
$\tilde{\Delta}_{i,j}(\boldsymbol{a})$	EXPECTED CLICK DIFFERENCE,$\tilde{\Delta}_{i,j}(\boldsymbol{a}) = \mathbb{E}_{\boldsymbol{a}' \sim \mathcal{U}(\{\boldsymbol{a},(i,j)\circ\boldsymbol{a}\})}\left[c_i - c_j \mid c_i \neq c_j\right]$
$\hat{s}_{i,j}(t)$	UNIRANK'S MAIN STATISTIC TO INFER THAT $i \succ j$, $\hat{s}_{i,j}(t) \overset{def}{=} \frac{1}{T_{i,j}(t)} \sum_{s=1}^{t-1} \mathbb{1}\left\{\exists c, (i,j) \in P_c(s)^2\right\} (c_i(s) - c_j(s))$
$\underline{s}_{i,j}(t)$	KULLBACK-LEIBLER BASED PESSIMISTIC ESTIMATOR, $\underline{s}_{i,j}(t) \overset{def}{=} 2 * f\left(\frac{1+\hat{s}_{i,j}(t)}{2}, T_{i,j}(t), \tilde{t}_{\tilde{\boldsymbol{P}}}(t)\right) - 1$
$\tilde{s}_{i,j}(t)$	SLIGHTLY PESSIMISTIC ESTIMATOR $\tilde{s}_{i,j}(t) \overset{def}{=} \hat{s}_{i,j}(t) - \sqrt{\log\log t / T_{i,j}(t)}$
f	KULLBACK-LEIBLER INDEX FUNCTION, $f(\hat{\mu}, T, t) \overset{def}{=} \inf\{\mu \in [0, \hat{\mu}] : T \times \mathrm{kl}(\hat{\mu}, \mu) \leq \log(t) + 3\log(\log(t))\}$,
$\mathrm{kl}(p,q)$	KULLBACK-LEIBLER DIVERGENCE FROM A BERNOULLI DISTRIBUTION OF MEAN p TO A BERNOULLI DISTRIBUTION OF MEAN q, $\mathrm{kl}(p,q) = p\log\left(\frac{p}{q}\right) + (1-p)\log\left(\frac{1-p}{1-q}\right)$
$\mathcal{U}(S)$	UNIFORM DISTRIBUTION ON THE SET S
c	(IN PB-MHB) PARAMETER CONTROLLING SIZE OF THE STEP IN THE METROPOLIS HASTING INFERENCE

Table B.1: Summary of the notations of Chapter 5.

Table B.1 summarizes the notations used throughout the paper and the appendix. Below are additional notations necessary for the proofs.

Definition 4 (Specific notations to count events and observations)**.** *The proofs are based on the concentration of the statistic $\hat{s}_{i,j}(t)$ which is the average over $T_{i,j}(t)$ observations. The number $T_{i,j}(t)$ itself is a sum: the sum of the random variables $\mathbb{1}\{c_i(s) \neq c_j(s)\}$ |*

$\exists c, (i,j) \in P_c(s)^2$, *where* s *is in* $[t]$. *To discuss the concentration of this sum, for any iteration* t *in* $[T]$, *we denote* $t_{i,j}(t) \overset{def}{=} \sum_{s=1}^{t-1} \mathbb{1}\left\{\exists c, (i,j) \in P_c(s)^2\right\}$ *the number of iterations at which the random variable is observed.*

Definition 5 (Recommended subset)**.** *Let* (L, K, ρ) *be an online learning to rank problem,* $\boldsymbol{P}$ *be an ordered partition of* $[L]$ *in* d *subsets, and* $c \in [d]$ *the index of one of these subsets. The subset* P_c *is* recommended *(denoted* $\mathrm{Rec}(P_c)$*) if the recommendations compatible with* $\boldsymbol{P}$ *include some items from* P_c. *More specifically, the subset* P_c *is* recommended *if* $\left|\bigcup_{\ell \in [c-1]} P_\ell\right| < K$.

Definition 6 (Expectations on clicks)**.** *let* i *and* j *be two different items.*

We denote

$$\tilde{\delta}_{i,j} \overset{def}{=} \min_{\boldsymbol{P}:\exists c,(i,j)\in P_c^2 \wedge \mathrm{Rec}(P_c)} \mathbb{P}_{\boldsymbol{a}(t)\sim\mathcal{U}(\mathcal{A}(\boldsymbol{P}))}\left[c_i(t) \neq c_j(t)\right]$$

the smallest probability for $c_i(t)$ *to be different from* $c_j(t)$ *while both items are in the same subset of the chosen partition* $\boldsymbol{P}(t)$ *(and may potentially be clicked upon). If we assume* $1 \succ 2 \succ \cdots \succ L$, *we also denote*

$$\tilde{\delta}_i^* \overset{def}{=} \min_{\boldsymbol{P}\in\mathcal{N}((\{1\},\ldots,\{K\},\{K+1,\ldots,L\})):\exists c,(\min(i-1,K),i)\in P_c^2} \mathbb{P}_{\boldsymbol{a}(t)\sim\mathcal{U}(\mathcal{A}(\boldsymbol{P}))}\left[c_{\min(i-1,K)}(t) \neq c_i(t)\right]$$

the smallest probability for $c_{\min(i-1,K)}(t)$ *to be different from* $c_i(t)$ *while both items* $\min(i-1, K)$ *and* i *are in the same subset of the chosen partition* $\boldsymbol{P}(t)$ *(and may potentially be clicked upon), and* $\boldsymbol{P}(t)$ *is in the neighborhood of the optimal partition* $\boldsymbol{P}^* = (\{1\}, \ldots, \{K\}, \{K+1, \ldots, L\})$.

If $i \succ j$, *we denote*

$$\tilde{\Delta}_{i,j} \overset{def}{=} \min_{\boldsymbol{P}:\exists c,(i,j)\in P_c^2 \wedge \mathrm{Rec}(P_c)} \mathbb{E}_{\boldsymbol{a}(t)\sim\mathcal{U}(\mathcal{A}(\boldsymbol{P}))}\left[c_i(t) - c_j(t) \mid c_i(t) \neq c_j(t)\right] = \min_{\boldsymbol{a}\in\mathcal{P}_K^L:\{i,j\}\cap\boldsymbol{a}([K])\neq\varnothing} \tilde{\Delta}_{i,j}(\boldsymbol{a}),$$

the smallest expected difference of clicks between items i *and* j *while both items are in the same subset of the chosen partition* $\boldsymbol{P}(t)$ *(and may potentially be clicked upon).*

Symmetrically, if $j \succ i$, *we denote*

$$\tilde{\Delta}_{i,j} \overset{def}{=} \max_{\boldsymbol{P}:\exists c,(i,j)\in P_c^2 \wedge \mathrm{Rec}(P_c)} \mathbb{E}_{\boldsymbol{a}(t)\sim\mathcal{U}(\mathcal{A}(\boldsymbol{P}))}\left[c_i(t) - c_j(t) \mid c_i(t) \neq c_j(t)\right] = \max_{\boldsymbol{a}\in\mathcal{P}_K^L:\{i,j\}\cap\boldsymbol{a}([K])\neq\varnothing} \tilde{\Delta}_{i,j}(\boldsymbol{a}),$$

the greatest expected difference of clicks between items i *and* j *while both items are in the same subset of the chosen partition* $\boldsymbol{P}(t)$ *(and may potentially be clicked upon).*

Lemma 7 in Appendix B.4.2 ensures the proper definition of these notations under Assumptions 2, 4, and 5, and states that $\tilde{\delta}_{i,j} = \tilde{\delta}_{j,i} > 0$ *and* $\tilde{\Delta}_{i,j} = -\tilde{\Delta}_{j,i} > 0$ *if* $i \succ j$.

Definition 7 (Reward gap)**.** *Let* (L, K, ρ) *be an OLR problem satisfying Assumption 4 and such that the order on items is a total order. Without loss of generality, let us assume that* $1 \succ 2 \succ \cdots \succ L$.. *Denoting* $\boldsymbol{P}^* = (\{1\}, \ldots, \{K\}, \{K+1, \ldots, L\})$ *the optimal partition associated to this order and taking* $c \geqslant 2$*, the* reward gap *of item* c *is*

$$\begin{aligned}\Delta_c \overset{def}{=} & \rho\left(\boldsymbol{a}^*, \min(c-1, K)\right) + \rho\left(\boldsymbol{a}^*, c\right) \\ & - \rho\left((\min(c-1, K), c) \circ \boldsymbol{a}^*, \min(c-1, K)\right) - \rho\left((\min(c-1, K), c) \circ \boldsymbol{a}^*, c\right)\end{aligned}$$

Note that for $c \leqslant K$, $\Delta_c \overset{def}{=} \rho(\boldsymbol{a}^*, c-1) + \rho(\boldsymbol{a}^*, c) - \rho((c-1, c) \circ \boldsymbol{a}^*, c-1) - \rho((c-1, c) \circ \boldsymbol{a}^*, c)$*, and for* $c \geqslant K+1$, $\Delta_c = \rho(\boldsymbol{a}^*, K) - \rho((K, c) \circ \boldsymbol{a}^*, K)$.

B.3 Proof of Lemma 4 (PBM and CM Fulfills Assumptions 2, 4, and 5)

For both CM and PBM click models, we note θ_i the click probability of item i. For PBM we have κ_k the probability that a user see the position k.

Proof. Let us begin with some preliminary remarks.

First, with PBM model, the positions are ranked by decreasing observation probability, meaning that $\kappa_{\boldsymbol{a}_1} \geqslant \kappa_{\boldsymbol{a}_2} \geqslant \cdots \geqslant \kappa_{\boldsymbol{a}_K}$.

Secondly, by definition, $\rho(k, \boldsymbol{a}) > 0$ for any position k and recommendation $\boldsymbol{a}$, which implies that:

— $\min_i \theta_i > 0$ and $\max_i \theta_i < 1$ in CM model;

— $\kappa_K > 0$ in PBM model.

Let us now prove that Assumptions 2, 4 and 5 are fulfilled by PBM and CM click models with the strict weak order $\succ$ defined by $i \succ j \iff \theta_i > \theta_j$.

By definition of $\succ$, Assumption 2 is fulfilled taking the the preferential attachment function $g : i \mapsto \theta_i$, and 3 is fulfilled as soon as $\theta_i \neq \theta_j$ for any items $i \neq j$

For Assumption 4, we have to prove that having $\boldsymbol{a}$ compatible with $\succ$ is optimal, meaning $\mu_{\boldsymbol{a}} = \mu^*$.

Let $\boldsymbol{a}$ be a permutation compatible with $\succ$.

In the case of CM, $\mu_{\boldsymbol{a}} = 1 - \sum_{k=1}^{K}(1-\theta_{a_k})$. In order to maximize $\mu_{\boldsymbol{a}}$, one has to select the K higher values of $\boldsymbol{\theta}$. As $\boldsymbol{a}$ is compatible with $\succ$, which is defined based on values θ_i, it satisfies this property. Hence, CM fulfills Assumption 4.

For PBM, $\mu_{\boldsymbol{a}} = \sum_{k=1}^{K}\theta_{\boldsymbol{a}_k}\kappa_k$. As the series $(\kappa_k)_{k\in[K]}$ is non-increasing, $\mu_{\boldsymbol{a}}$ is maximized if $(\theta_k)_{k\in[K]}$ is also non-increasing and if $\theta_K \geqslant \max_{k\leqslant K+1}\theta_k$. These properties are ensured by the fact that $\boldsymbol{a}$ is compatible with $\succ$ and that $\succ$ is defined based on values θ_i. Hence, PBM fulfills Assumption 4.

We now prove that CM and PBM fulfill Assumption 5. Let i and j be two distinct items such that $i \succ j$ and $\boldsymbol{a} \in \mathcal{P}_K^L$ be a recommendation such that at least one of both items is displayed.

First, $\mathbb{E}_{\boldsymbol{a}'\sim\mathcal{U}(\{\boldsymbol{a},(i,j)\circ\boldsymbol{a}\})}\left[c_i(t) \neq c_j(t) \mid \boldsymbol{a}(t) = \boldsymbol{a}'\right]$ is non-null with PBM model as $c_i(t)$ and $c_j(t)$ are independent and as at least one of the four variables $c_i(t) \mid \boldsymbol{a}(t) = \boldsymbol{a}$, $c_i(t) \mid \boldsymbol{a}(t) = (i,j)\circ\boldsymbol{a}$, $c_j(t) \mid \boldsymbol{a}(t) = \boldsymbol{a}$, $c_j(t) \mid \boldsymbol{a}(t) = (i,j)\circ\boldsymbol{a}$ has an expectation which is non-zero and strictly smaller than 1 (due to $\kappa_K > 0$ and $\theta_i > \theta_j$).

Similarly, $\mathbb{E}_{\boldsymbol{a}'\sim\mathcal{U}(\{\boldsymbol{a},(i,j)\circ\boldsymbol{a}\})}\left[c_i(t) \neq c_j(t) \mid \boldsymbol{a}(t) = \boldsymbol{a}'\right]$ is non-null with CM model as at most one of both items can be clicked at each iteration and the shown item has non-zero probability to be clicked (by definition of ρ).

Then, we consider $\tilde{\Delta}_{i,j}(\boldsymbol{a})$ as

$$\tilde{\Delta}_{i,j}(\boldsymbol{a}) = \frac{\mathbb{P}_{\boldsymbol{a}'\sim\mathcal{U}(\{\boldsymbol{a},(i,j)\circ\boldsymbol{a}\})}(c_i = 1, c_j = 0) - \mathbb{P}_{\boldsymbol{a}'\sim\mathcal{U}(\{\boldsymbol{a},(i,j)\circ\boldsymbol{a}\})}(c_i = 0, c_j = 1)}{\mathbb{P}_{\boldsymbol{a}'\sim\mathcal{U}(\{\boldsymbol{a},(i,j)\circ\boldsymbol{a}\})}(c_i = 1, c_j = 0) + \mathbb{P}_{\boldsymbol{a}'\sim\mathcal{U}(\{\boldsymbol{a},(i,j)\circ\boldsymbol{a}\})}(c_i = 0, c_j = 1)}$$

We want to control the sign of $\tilde{\Delta}_{i,j}(\boldsymbol{a})$, which is also the sign of its numerator, as its denominator (noted $D_{\tilde{\Delta}_{i,j}(\boldsymbol{a})}$) is non-negative.

The recommendation $\boldsymbol{a}'$ is drawn uniformly in $\{\boldsymbol{a}, (i,j)\circ\boldsymbol{a}\}$ thus

$$\mathbb{P}_{\boldsymbol{a}'\sim\mathcal{U}(\{\boldsymbol{a},(i,j)\circ\boldsymbol{a}\})}(c_i = 1, c_j = 0) = \frac{1}{2}\mathbb{P}_{\boldsymbol{a}}(c_i = 1, c_j = 0) + \frac{1}{2}\mathbb{P}_{(i,j)\circ\boldsymbol{a}}(c_i = 1, c_j = 0).$$

When considering a CM click model, we have $\mathbb{P}_{\boldsymbol{a}}(c_i = 1, c_j = 0) = \prod_{p=1}^{k-1}(1-\theta_{a_p})\theta_i$ and $\mathbb{P}_{\boldsymbol{a}}(c_i = 0, c_j = 1) = \prod_{p=1}^{l-1}(1-\theta_{a_p})\theta_j$ when i and j $\in \boldsymbol{a}$.

In that case, we have:

$$\tilde{\Delta}_{i,j}(\boldsymbol{a}) = \frac{\frac{1}{2}\prod_{p=1}^{k-1}(1-\theta_{a_p})\theta_i + \frac{1}{2}\prod_{p=1}^{l-1}(1-\theta_{a_p})\theta_i - \left(\frac{1}{2}\prod_{p=1}^{l-1}(1-\theta_{a_p})\theta_j + \frac{1}{2}\prod_{p=1}^{k-1}(1-\theta_{a_p})\theta_j\right)}{D_{\tilde{\Delta}_{i,j}(\boldsymbol{a})}}$$

which can be simplified in:

$$\tilde{\Delta}_{i,j}(\boldsymbol{a}) = \frac{\frac{1}{2}\left(\prod_{p=1}^{k-1}(1-\theta_{a_p}) + \prod_{p=1}^{l-1}(1-\theta_{a_p})\right)(\theta_i - \theta_j)}{D_{\tilde{\Delta}_{i,j}(\boldsymbol{a})}}.$$

Since $\max_i \theta_i < 1$, $\prod_{p=1}^{k-1}(1-\theta_{a_p}) + \prod_{p=1}^{l-1}(1-\theta_{a_p}) > 0$, thus the sign of $\tilde{\Delta}_{i,j}(\boldsymbol{a})$ is the sign of $(\theta_i - \theta_j)$ and $\tilde{\Delta}_{i,j}(\boldsymbol{a}) > 0 \iff \theta_i > \theta_j \iff i \succ j$.

Now if $i \notin \boldsymbol{a}$ then $\mathbb{P}_{\boldsymbol{a}}(c_i = 1, c_j = 0) = 0$ as the position is not seen. We have:

$$\tilde{\Delta}_{i,j}(\boldsymbol{a}) = \frac{\frac{1}{2}(\prod_{p=1}^{l-1}(1-\theta_{a_p}))(\theta_i - \theta_j)}{D_{\tilde{\Delta}_{i,j}(\boldsymbol{a})}}$$

which leads to the same conclusion as the previous case. By symmetry, we have the same conclusion with j $\notin \boldsymbol{a}$.

Now with a PBM click model, we have $\mathbb{P}_{\boldsymbol{a}}(c_i = 1, c_j = 0) = \kappa_k\theta_i(1 - \kappa_l\theta_j)$ as $c_i = 1$ and $c_j = 0$ are independant events.

Thus, we have:

$$\tilde{\Delta}_{i,j}(\boldsymbol{a}) = \frac{\frac{1}{2}\kappa_k\theta_i(1-\kappa_l\theta_j) + \frac{1}{2}\kappa_l\theta_i(1-\kappa_k\theta_j) - \left(\frac{1}{2}\kappa_l\theta_j(1-\kappa_k\theta_i) + \frac{1}{2}\kappa_k\theta_j(1-\kappa_l\theta_i)\right)}{D_{\tilde{\Delta}_{i,j}(\boldsymbol{a})}}$$

which can be simplified in:

$$\tilde{\Delta}_{i,j}(\boldsymbol{a}) = \frac{\frac{1}{2}(\kappa_k + \kappa_l)(\theta_i - \theta_j)}{D_{\tilde{\Delta}_{i,j}(\boldsymbol{a})}}$$

As κ_k or κ_l is positive if i or j is presented, similarly to the CM case we have $\tilde{\Delta}_{i,j}(\boldsymbol{a}) > 0 \iff \theta_i > \theta_j \iff i \succ j$.

This proof can be extended to i or $j \notin \boldsymbol{a}$ by taking $\kappa_k = 0$ when $k > K$.

We can conclude that both CM and PBM fulfills Assumption 5. □

B.4 Technical Lemmas Required by the Proof of Theorem 3

In this section, we gather technical Lemmas required to prove the regret upper-bound of UniRank. These lemmas regard the pseudo-unimodality of the considered setting (Ap-

pendix B.4.1), the concentration away from zero of the statistic $\hat{s}_{i,j}(t)$ (Appendices B.4.2 and B.4.3), and the sufficient optimism brought by $\underline{s}_{i,j}(t)$ (Appendix B.4.4).

B.4.1 Proof of Lemma 5 (Pseudo-Unimodality Assuming a Total Order on Items)

Proof. To ease the notations, we take the following order on items: $1 \succ 2 \succ \cdots \succ L$. Therefore, $\boldsymbol{P}^* = (\{1\}, \ldots, \{K\}, \{j \in [L] \setminus [K]\})$. Thus $\tilde{\boldsymbol{P}} \neq \boldsymbol{P}^*$ implies that $\tilde{\boldsymbol{P}}$ does not have the same attributes as $\boldsymbol{P}^*$:

— either there exists $c \in [d-1]$ such that $|\tilde{P}_c| > 1$;

— or $\tilde{\boldsymbol{P}} = (\{1\}, \ldots, \{K\}, \{\ell : \ell \in [L] \setminus [K]\})$, with $\bigcup_{\ell=1}^{L} \{\ell\} = [L]$, there exists c in $[K]$, such that $\tilde{P}_c = \{i\}$, and there exists $j \in \tilde{P}_{c+1}$ such that $j \succ i$.

Let us show that this second alternative is divided into the two last outputs of Lemma 5. Let $c \in [K]$ be the smallest index such that $\tilde{P}_c = \{i\}$ and there exists $j \in \tilde{P}_{c+1}$ such that $j \succ i$. Either $c > 1$, and therefore $\tilde{P}_{c-1} = \{i'\}$ and $i' \succ i$, or $c = 1$. □

B.4.2 Minimum Expected Click Difference

Assumption 5 builds upon $\tilde{\Delta}_{i,j}(\boldsymbol{a})$ which measures the difference of attractiveness between i and j while all other items are at fixed positions. In the theoretical analysis of UniRank, we handle situations where other items may also change in position thanks to the following Lemma.

Lemma 7 (Minimum expected click difference)**.** *Let (L, K, ρ) be an OLR problem satisfying Assumptions 4 and 5 with $\succ$ the order on items, and let i and j be two items such that $i \succ j$. Then, for any partition of items $\boldsymbol{P}$, if there exists c such that $(i,j) \in P_c^2$ and $\mathbb{E}_{\boldsymbol{a}(t) \sim \mathcal{U}(\mathcal{A}(\boldsymbol{P}))}[c_i(t) \neq c_j(t)] \neq 0$, then $\mathbb{E}_{\boldsymbol{a}(t) \sim \mathcal{U}(\mathcal{A}(\boldsymbol{P}))}[c_i(t) - c_j(t) \mid c_i(t) \neq c_j(t)] > 0$ and therefore*

$$\tilde{\delta}_{i,j} > 0 \qquad \text{and} \qquad \tilde{\Delta}_{i,j} > 0.$$

Symmetrically, if $j \succ i$, for any partition of items $\boldsymbol{P}$, if there exists c such that $(i,j) \in P_c^2$ and $\mathbb{E}_{\boldsymbol{a}(t) \sim \mathcal{U}(\mathcal{A}(\boldsymbol{P}))}[c_i(t) \neq c_j(t)] \neq 0$, then $\mathbb{E}_{\boldsymbol{a}(t) \sim \mathcal{U}(\mathcal{A}(\boldsymbol{P}))}[c_i(t) - c_j(t) \mid c_i(t) \neq c_j(t)] < 0$ and therefore

$$\tilde{\delta}_{i,j} > 0 \qquad \text{and} \qquad \tilde{\Delta}_{i,j} < 0.$$

Proof. The proof consists in writing $\mathbb{E}_{\boldsymbol{a}(t)\sim\mathcal{U}(\mathcal{A}(\boldsymbol{P}))}\left[c_i(t) \neq c_j(t)\right] \neq 0$ two times as a sum other $\boldsymbol{a}(t) \in \mathcal{U}(\mathcal{A}(\boldsymbol{P}))$, and in reindexing one of both sums by $(i,j) \circ \boldsymbol{a}(t) \in \mathcal{U}(\mathcal{A}(\boldsymbol{P}))$. Then, adding the terms of both sums we get a sum of terms $\tilde{\Delta}_{i,j}(\boldsymbol{a})$ which by assumption 5 are positive. Hence this sum is positive, which concludes the proof. □

B.4.3 Upper-bound on the Number of High Deviations for Variables with Lower-Bounded Mean

The Proof of Theorem 3 requires the control of the expected number of high deviations of the statistic $\hat{s}_{i,j}(t)$. We control this expectation through Lemma 10 which derives from the application of Lemmas 8 and 9 to $\hat{s}_{i,j}(t)$ and $\hat{T}_{i,j}(t)$. Hereafter, we express and prove the three lemmas. Note that Lemmas 8 and 9 are extensions of Lemmas 4.3 and B.1 of [13] to a setting where the handled statistic is a mixture of variables following different laws of bounded expectation.

Lemma 8 (Concentration bound with lower-bounded mean)**.** *Let $(X_t^a)_{t\geqslant 1}$ with $a \in \mathcal{R}$, be $|\mathcal{R}| < \infty$ independent sequences of independent random variables bounded in $[0,B]$ defined on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$. Let $\mathcal{F}_t$ be an increasing sequence of $\sigma-$fields of $\mathcal{F}$ such that for each t, $\sigma((X_1^a)_{a\in\mathcal{R}}, \ldots, (X_t^a)_{a\in\mathcal{R}}) \subset \mathcal{F}_t$ and for $s > t$ and a a recommendation, X_s^a is independent from $\mathcal{F}_t$. Consider $|\mathcal{R}|$ previsible sequences $(\epsilon_t^a)_{t\geq 1}$ of Bernoulli variables (for all $t > 0$, ϵ_t^a is $\mathcal{F}_{t-1} -$ mesurable) such that for all $t > 0$, $\sum_i \epsilon_t^a \in \{0,1\}$. Let $\delta > 0$ and for every $t \in \{1, \ldots, n\}$ let*

$$S(t) = \sum_{s=1}^{t}\sum_i \epsilon_s^i (X_s^i - \mathbb{E}[X_s^i]), \qquad T(t) = \sum_{s=1}^{t}\sum_i \epsilon_s^i, \qquad \hat{\mu}(t) = \frac{S(t)}{N(t)}.$$

Define $\phi \in \{t_0, \ldots, T+1\}$ a $\mathcal{F}$-stopping time such that either $T(\phi) \geqslant s$ or $\phi = T+1$. Then

$$\mathbb{P}\left(S(\phi) \geqslant T(\phi)\delta, \phi \leqslant T\right) \leqslant \exp(-\frac{2n\delta^2}{B^2}).$$

Proof. Let $\lambda > 0$, and define $G_t = \exp(\lambda(S(t) - \delta T(t)))\mathbb{1}\{t \leqslant T\}$. We have that:

$$\begin{aligned}
\mathbb{P}(S(\phi) \geqslant T(\phi)\delta, \phi \leqslant T) &= \mathbb{P}(\exp(\lambda(S(\phi) - \delta T(\phi))\mathbb{1}\{\phi \leqslant T\} \geqslant 1) \\
&= \mathbb{P}(g_\phi \geqslant 1) \\
&\leqslant \mathbb{E}[G_\phi].
\end{aligned}$$

Next we provide an upper bound for $\mathbb{E}[G_\phi]$. We define the following quantities:

$$Y_s^i = \varepsilon_s^i(\lambda(X_s^i - \mathbb{E}[X_s^i]) - \lambda^2 B^2/8)$$

$$\tilde{G}_t = \exp\left(\sum_{s=1}^{t}\sum_i Y_s^i\right)\mathbb{1}\{t \leqslant T\}.$$

Taking $\lambda = 4\delta/B^2$, G_t can be written:

$$G_t = \tilde{G}_t \exp(-T(t)(\lambda\delta - \lambda^2 B^2/8) = \tilde{G}_t \exp(-2T(t)\delta^2/B^2).$$

As $T(t) \geqslant n$ if $\phi \leqslant T$ we can upper bound G_ϕ by:

$$G_\phi = \tilde{G}_\phi \exp(-2T(\phi)\delta^2/B^2) \leqslant \tilde{G}_\phi \exp(-2n\delta^2/B^2).$$

It is noted that the above inequality holds even when $\phi = T+1$, since $G_{T+1} = \tilde{G}_{T+1} = 0$. Hence:

$$\mathbb{E}[G_\phi] \leqslant \mathbb{E}[\tilde{G}_\phi]\exp(-2n\delta^2/B^2)$$

We prove that $\left(\tilde{G}_t\right)_t$ is a super-martingale. We have that $\mathbb{E}[\tilde{G}_{T+1} \mid \mathcal{F}_T] = 0 \leqslant \tilde{G}_T$. For $s \leqslant T-1$, since B_{t+1} is $\mathcal{F}$ measurable:

$$\mathbb{E}[\tilde{G}_{t+1} \mid \mathcal{F}_t] = \tilde{G}_t((1 - \sum_i \varepsilon_{t+1}^i) + \sum_i \varepsilon_{t+1}^i \mathbb{E}[\exp(Y_{t+1}^i)]).$$

As proven in (Hoeffding, 1963)[eq. 4.16] since $X_{t+1}^i \in [0, B]$:

$$\mathbb{E}[\exp(\lambda(X_{t+1}^i - \mathbb{E}[X_{t+1}^i])] \leqslant \exp(\lambda^2 B^2/8),$$

so $EE[\exp(Y_{t+1}^i)] \leqslant 1$ and $\left(\tilde{G}_t\right)_t$ is a super-martingale: $\mathbb{E}[\tilde{G}_{t+1} \mid \mathcal{F}_t] \leqslant \tilde{G}_t$. Since $\phi \leqslant T+1$ almost surely, and $\left(\tilde{G}_t\right)_t$ is a supermartingale, Doob's optional stopping theorem yields: $\mathbb{E}[\tilde{G}_\phi] \leqslant \mathbb{E}[\tilde{G}_0] = 1$, and so

$$\begin{aligned}
\mathbb{P}(S(\phi) \geqslant T(\phi)\delta, \phi \leqslant T) &\leqslant \mathbb{E}[G_\phi] \\
&\leqslant \mathbb{E}[\tilde{G}_\phi]\exp(-2n\delta^2/B^2) \\
&\leqslant \exp(-2n\delta^2/B^2),
\end{aligned}$$

which concludes the proof □

Lemma 9 (Expected number of large deviation with lower-bounded mean)**.** *Let (L, K, ρ) be an OLR problem, $\mathcal{F}_t$ the natural σ-algebra generated by the OLR problem, and $\mathcal{F} = (\mathcal{F}_t)_{t\in\mathbb{Z}}$ the corresponding filtration. We denote $O_t \overset{def}{=} (\boldsymbol{a}(1), \boldsymbol{c}(1), \ldots, \boldsymbol{a}(t-1), \boldsymbol{c}(t-1))$ the set of random values observed up to time $t-1$. Let $Z_t \in [0, B]$ and $B_t \in \{0, 1\}$ be two $\mathcal{F}_{t-1}$-measurable random variables, $\Lambda \subseteq \mathbb{N}$ be a random set of instants, and $\varepsilon > 0$. For any $t \in \mathbb{Z}$, we denote $S(t) \overset{def}{=} \sum_{s=0}^{t} B_s Z_s$ and $T(t) \overset{def}{=} \sum_{s=0}^{t} B_s$. If for any $t > 0$, $\mathbb{E}[Z_t \mid 0_t, B_t = 1] \geqslant \delta$ and there exists a sequence of random sets $(\Lambda(n))_{n>0}$ such that (i) $\Lambda \subseteq \bigcup_{n>0} \Lambda(n)$, (ii) for all $n > 0$ and all $t \in \Lambda(n)$, $T(t) \geqslant \varepsilon n$, (iii) $|\Lambda(n)| \leqslant 1$, and (iv) the event $t \in \Lambda(n)$ is $\mathcal{F}$-measurable. Then*

$$\mathbb{E}\left[\sum_{t\geq 1} \mathbb{1}\{t \in \Lambda : S(t) < \frac{\delta}{2} T(t)\}\right] \leq \frac{2B^2}{\epsilon\delta^2}$$

Proof. Let $T \in \mathbb{N}$. For all $n \in \mathbb{N}$, $|\Lambda(n)| \leqslant 1$, we define Φ_n as $T+1$ if $\Lambda(n) \cap [T]$ is empty and $\{\Phi_n\} = \Lambda(n)$ otherwise. Since $\Lambda \subseteq \bigcup_{n>0} \Lambda(n)$, we have

$$\sum_{t=1}^{T} \mathbb{1}\left\{t \in \Lambda : S(t) < \frac{\delta}{2} T(t)\right\} \leqslant \sum_{n\geqslant 1} \mathbb{1}\left\{S(\Phi_n) < \frac{\delta}{2} T(\Phi_n), \Phi_n \leqslant T\right\}.$$

Taking expectations,

$$\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\left\{t \in \Lambda : S(t) < \frac{\delta}{2} T(t)\right\}\right] \leqslant \sum_{n\geqslant 1} \mathbb{P}\left[S(\Phi_n) < \frac{\delta}{2} T(\Phi_n), \Phi_n \leqslant T\right]$$

For any $t \in \mathbb{N}$, denote $S'(t) \overset{def}{=} \sum_{s=0}^{t} B_s(Z_s - \mathbb{E}[Z_s \mid 0_s, B_s = 1])$. As for any $s \in \mathbb{N}$, $\mathbb{E}[Z_s \mid 0_s, B_s = 1] > \delta$, $S'(t) < S(t) - T(t)\delta$. Therefore, for any $n \in \mathbb{N}$

$$\mathbb{P}\left[S(\Phi_n) < \frac{\delta}{2} T(\Phi_n), \Phi_n \leqslant T\right] \leqslant \mathbb{P}\left[S'(\Phi_n) < -\frac{\delta}{2} T(\Phi_n), \Phi_n \leqslant T\right]$$

and

$$\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\left\{t \in \Lambda : S(t) < \frac{\delta}{2} T(t)\right\}\right] \leqslant \sum_{n\geqslant 1} \mathbb{P}\left[S'(\Phi_n) < -\frac{\delta}{2} T(\Phi_n), \Phi_n \leqslant T\right]$$

By Lemma 8, since Φ_n is a stopping time upper bounded by $T+1$, and $T(\Phi_n) \geqslant \varepsilon n$,

$$\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\left\{t \in \Lambda : S(t) < \frac{\delta}{2} T(t)\right\}\right] \leqslant \sum_{n\geqslant 1} \exp\left(-\frac{\varepsilon n \delta^2}{2B^2}\right) \leqslant \frac{2B^2}{\varepsilon\delta^2},$$

where the last inequality drives from the $\sum_{n\geqslant 1} \exp(-nw) \leqslant \int_0^{+\infty} \exp(-uw)\, du = \frac{1}{w}$.

This upper-bound is valid for any T, which concludes the proof. □

Lemma 10 (Expected number of large deviation for our statistics)**.** *Let (L, K, ρ) be an OLR problem satisfying Assumptions 4 and 5 with $\succ$ the order on items, and let i and j be two items. If there exists a sequence of random sets $(\Lambda(n))_{n>0}$ such that (i) $\Lambda \subseteq \bigcup_{n>0} \Lambda(n)$, (ii) for all $n > 0$ and all $t \in \Lambda(n)$, $t_{i,j}(t) \geqslant \varepsilon n$, (iii) $|\Lambda(n)| \leqslant 1$, and (iv) the event $t \in \Lambda(n)$ is $\mathcal{F}$-measurable. Then,*

$$\mathbb{E}\left[\sum_{t\geq 1} \mathbb{1}\left\{t \in \Lambda, T_{i,j}(t) < \frac{\tilde{\delta}_{i,j}}{2} t_{i,j}(t)\right\}\right] = \mathcal{O}(1) \tag{B.1}$$

and

$$\mathbb{E}\left[\sum_{t\geq 1} \mathbb{1}\left\{t \in \Lambda, \frac{\hat{s}_{i,j}(t)}{\tilde{\Delta}_{i,j}} < \frac{1}{2}\right\}\right] = \mathcal{O}(1),$$

meaning

$$\mathbb{E}\left[\sum_{t\geq 1} \mathbb{1}\left\{t \in \Lambda, \hat{s}_{i,j}(t) < \frac{\tilde{\Delta}_{i,j}}{2}\right\}\right] = \mathcal{O}(1) \qquad \text{, if } i \succ j; \tag{B.2}$$

$$\mathbb{E}\left[\sum_{t\geq 1} \mathbb{1}\left\{t \in \Lambda, \hat{s}_{i,j}(t) > \frac{\tilde{\Delta}_{i,j}}{2}\right\}\right] = \mathcal{O}(1) \qquad \text{, if } j \succ i. \tag{B.3}$$

Proof. Let assume $i \succ j$. We first prove Claim (B.1) and then prove Claim (B.2) using Claim (B.1).

For any $t \leqslant 1$, we define both following $\mathcal{F}_{t-1}$-measurable random variables

$$Z_t \overset{def}{=} \mathbb{1}\left\{c_i(t) \neq c_j(t)\right\} \qquad\qquad B_t \overset{def}{=} \mathbb{1}\left\{\exists c, (i,j) \in P_c(t)^2\right\},$$

and we denote $O_t \overset{def}{=} (\boldsymbol{a}(1), \boldsymbol{c}(1), \ldots, \boldsymbol{a}(t-1), \boldsymbol{c}(t-1))$ the set of random values observed up to time $s-1$. Note that $T_{i,j}(t+1) = \sum_{s=1}^{t} B_s Z_s$, $t_{i,j}(t+1) = \sum_{s=1}^{t} B_s$, and $\mathbb{E}\left[Z_t \mid 0_t, B_t = 1\right] > \tilde{\delta}_{i,j}$ by Lemma 7.

Therefore by Lemma 9

$$\mathbb{E}\left[\sum_{t\geq 1} \mathbb{1}\{t \in \Lambda : T_{i,j}(t+1) < \frac{\tilde{\delta}_{i,j}}{2} t_{i,j}(t+1)\}\right] \leqslant \frac{2}{\epsilon \tilde{\delta}_{i,j}^2},$$

meaning

$$\mathbb{E}\left[\sum_{t\geq 1}\mathbb{1}\{t\in\Lambda : T_{i,j}(t) < \frac{\tilde{\delta}_{i,j}}{2}t_{i,j}(t)\}\right] \leqslant 1 + \frac{2}{\epsilon\tilde{\delta}_{i,j}^2} = \mathcal{O}(1),$$

which corresponds to Claim (B.1).

Let now prove Claim (B.2) using the following decomposition

$$\mathbb{E}\left[\sum_{t=1}^{T}\mathbb{1}\left\{t\in\Lambda, \hat{s}_{i,j}(t) < \frac{\tilde{\Delta}_{i,j}}{2}\right\}\right] \leqslant \mathbb{E}\left[\sum_{t=1}^{T}\mathbb{1}\left\{t\in\Lambda, \hat{s}_{i,j}(t) < \frac{\tilde{\Delta}_{i,j}}{2}T_{i,j}(t) < \frac{\tilde{\delta}_{i,j}}{2}t_{i,j}(t)\right\}\right] + \mathbb{E}\left[\sum_{t=1}^{T}\mathbb{1}\left\{t\in\Lambda, \hat{s}_{i,j}(t) < \frac{\tilde{\Delta}_{i,j}}{2}, T_{i,j}(t) \geqslant \frac{\tilde{\delta}_{i,j}}{2}t_{i,j}(t)\right\}\right],$$

Where the first right-hand side term is smaller than $\mathbb{E}\left[\sum_{t\geq 1}\mathbb{1}\left\{t\in\Lambda, T_{i,j}(t) < \frac{\tilde{\delta}_{i,j}}{2}t_{i,j}(t)\right\}\right]$ and therefore is a $\mathcal{O}(1)$. We control the second term by applying again Lemma 9.

For any $t \leqslant 1$, we define both following $\mathcal{F}_{t-1}$-measurable random variables

$$Z_t \stackrel{def}{=} c_i(t) - c_j(t) \qquad\qquad B_t \stackrel{def}{=} \mathbb{1}\left\{\exists c, (i,j)\in P_c(t)^2, c_i(t)\neq c_j(t)\right\},$$

Note that $Z_t \in [-1,1]$, $\hat{s}_{i,j}(t+1)T_{i,j}(t+1) = \sum_{s=1}^{t} B_s Z_s$, $T_{i,j}(t+1) = \sum_{s=1}^{t} B_s$, and $\mathbb{E}\left[Z_t \mid 0_t, B_t = 1\right] > \tilde{\Delta}_{i,j}$ by Lemma 7 as $i \succ j$.

We also define $A \stackrel{def}{=} \Lambda \cap \left\{t\in\mathbb{N} : T_{i,j}(t) \geqslant \frac{\tilde{\delta}_{i,j}}{2}t_{i,j}(t)\right\}$ and for any $n\in\mathbb{N}$, $A(n) \stackrel{def}{=} \Lambda(n) \cap \left\{t\in\mathbb{N} : T_{i,j}(t) \geqslant \frac{\tilde{\delta}_{i,j}}{2}t_{i,j}(t)\right\}$. Then, (i) as $\Lambda \subseteq \bigcup_{n>0}\Lambda(n)$, $A \subseteq \bigcup_{n>0}A(n)$, (ii) for all $n>0$ and all $t\in A(n)$, $T_{i,j}(t) \geqslant \frac{\tilde{\delta}_{i,j}}{2}t_{i,j}(t) \geqslant \frac{\tilde{\delta}_{i,j}}{2}\varepsilon n$, (iii) $|A(n)| \leqslant |\Lambda(n)| \leqslant 1$, and (iv) the event $t\in A(n)$ is $\mathcal{F}$-measurable. Therefore by Lemma 9

$$\mathbb{E}\left[\sum_{t\geq 1}\mathbb{1}\{t\in A : \hat{s}_{i,j}(t+1)T_{i,j}(t+1) < \frac{\tilde{\Delta}_{i,j}}{2}T_{i,j}(t+1)\}\right] \leqslant \frac{8}{\tilde{\delta}_{i,j}\varepsilon\tilde{\Delta}_{i,j}^2},$$

meaning

$$\mathbb{E}\left[\sum_{t\geq 1}\mathbb{1}\left\{\begin{matrix} t\in\Lambda, \hat{s}_{i,j}(t) < \frac{\tilde{\Delta}_{i,j}}{2}, \\ T_{i,j}(t) \geqslant \frac{\tilde{\delta}_{i,j}}{2}t_{i,j}(t)\end{matrix}\right\}\right] \leqslant 1 + \frac{8}{\tilde{\delta}_{i,j}\varepsilon\tilde{\Delta}_{i,j}^2} = \mathcal{O}(1).$$

Overall, $\mathbb{E}\left[\sum_{t=1}^{T}\mathbb{1}\left\{t\in\Lambda, \hat{s}_{i,j}(t) < \frac{\tilde{\Delta}_{i,j}}{2}\right\}\right] = \mathcal{O}(1)+\mathcal{O}(1) = \mathcal{O}(1)$ which corresponds to Claim (B.2).

Other claims are proved symmetrically. □

B.4.4 Upper-Bound on the Number of Upper-Estimations of a Pessimistic Estimator

This section presents two results aiming at upper-bounding the number of iterations at which $\tilde{\Delta}_{i,j}$ is upper-estimated by $\underline{s}_{i,j}(t)$ if $i \succ j$. These new results are extensions of Lemma 9 and Theorem 10 of [21] to a setting where the handled statistic is a mixture of variables following different laws of bounded expectation.

Lemma 11. *Let X be a random variable taking value in $[0,1]$ and let $\mu \leq \mathbb{E}[X]$. then for all $\lambda < 0$,*

$$\mathbb{E}[\exp(\lambda X)] \leq 1 - \mu + \mu \exp(\lambda),$$

Proof. The function $\mathrm{f} : [0,1] \xrightarrow{\mathbb{R}}$ defined by $f(x) = \exp(\lambda x) - x(\exp(\lambda) - 1) - 1$ is convex and such that $f(0) = f(1) = 0$, hence $f(x) \leq 0$ for all $x \in [0,1]$. Consequentely,

$$\mathbb{E}[\exp(\lambda X)] \leq \mathbb{E}[X(\exp(\lambda) - 1) + 1] = \mathbb{E}[X](\exp(\lambda) - 1) + 1$$

As $\lambda < 0$ and $\mu \leq \mathbb{E}[X]$, we have $\mathbb{E}[X](\exp(\lambda) - 1) \leq \mu(\exp(\lambda) - 1)$ and

$$\mathbb{E}[\exp(\lambda X)] \leq \mu(\exp(\lambda) - 1) + 1$$

□

Lemma 12. *Let $(X_t^a)_{t \geqslant 1}$ with $a \in \mathcal{R}$, be $|\mathcal{R}| < \infty$ independent sequences of independent random variables bounded in $[0,1]$ defined on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ with common expectations $\mu^a = \mathbb{E}[X_t^a]$ of minimal value $\mu = \min_{a \in \mathcal{R}} \mu^a$. Let $\mathcal{F}_t$ be an increasing sequence of $\sigma-$fields of $\mathcal{F}$ such that for each t, $\sigma((X_1^a)_{a \in \mathcal{R}}, \ldots, (X_t^a)_{a \in \mathcal{R}}) \subset \mathcal{F}_t$ and for $s > t$ and a a recommendation, X_s^a is independent from $\mathcal{F}_t$. Consider $|\mathcal{R}|$ previsible sequences $(\epsilon_t^a)_{t \geq 1}$ of Bernoulli variables (for all $t > 0$, ϵ_t^a is $\mathcal{F}_{t-1} -$ mesurable) such that for all $t > 0$, $\sum_i \epsilon_t^a \in \{0,1\}$. Let $\delta > 0$ and for every $t \in \{1, \ldots, n\}$ let*

$$S(t) = \sum_{s=1}^{t} \sum_i \epsilon_s^i X_s^i, \qquad N(t) = \sum_{s=1}^{t} \sum_i \epsilon_s^i, \qquad \hat{\mu}(t) = \frac{S(t)}{N(t)}$$

$$u(n) = \max\{q > \hat{\mu}_n : N(n) d(\hat{\mu}(n), q) \leq \delta\}$$

Then

$$\mathbb{P}(u(n) < \mu) \leq e \lceil \delta \log(n) \rceil \exp(-\delta)$$

Proof. For every $\lambda < 0$, by Lemma 11, it holds that $\log(\mathbb{E}[\exp(\lambda X_1^a)]) \leq \log(1-\mu+\mu\exp(\lambda)) = \phi_\mu(\lambda)$ for all a. Let $W_0^\lambda = 1$ and for $t \geq 1$,

$$W_t^\lambda = \exp(\lambda S(t) - N(t)\phi_\mu(\lambda))$$

$(W_t^\lambda)_{t\geq}0$ is a super-martingale relative to $(\mathcal{F}_t)_{t\geq 0}$. In fact,

$$\mathbb{E}[\exp(\lambda\{S(t+1)-S(t)\})|\mathcal{F}_t] = \mathbb{E}[\exp(\lambda\sum_i \epsilon_{t+1}^i X_{t+1}^i)|\mathcal{F}_t]$$

As $(X_t^i)_t$ are independent sequences, we can rewrite :

$$\mathbb{E}[\exp(\lambda\{S(t+1)-S(t)\})|\mathcal{F}_t] = \prod_i \mathbb{E}[\exp(\lambda\epsilon_{t+1}^i X_{t+1}^i)|\mathcal{F}_t] = \prod_i \exp(\epsilon_{t+1}^i \log(\mathbb{E}[\exp(\lambda X_{t+1}^i)|\mathcal{F}_t]))$$

$$= \exp(\sum_i \epsilon_{t+1}^i \log(\mathbb{E}[\exp(\lambda X_1^i)|\mathcal{F}_t])) \leq \exp(\sum_i \epsilon_{t+1}^i \phi_\mu(\lambda)) = \exp(\{N(t+1)-N(t)\}\phi_\mu(\lambda))$$

which can be rewritten as

$$\mathbb{E}[\exp(\lambda S(t+1) - N(t+1)\phi_\mu(\lambda))|\mathcal{F}_t] \leq \exp(\lambda S(t) - N(t)\phi_\mu(\lambda))$$

The rest of the proof follows [21]. Using the "peeling trick", interval $\{1,\ldots,n\}$ of possible values for $N(n)$ is divided into slices $\{t_{k-1}+1,\ldots,t_k\}$ of geometrically increasing size. Each slice is treated independently. We assume that $\delta > 1$ and we construct the slicing as follow : $t_0 = 0$ and for $k \in \mathbb{N}^*$, $t_k = \lfloor(1+\eta)^k\rfloor$, with $\eta = 1/(\delta-1)$. Let$D = \lceil \frac{\log n}{\log 1+\eta}\rceil$ be the first inter such that $t_D \geq n$ and $A_k = \{t_{k-1} \leq N(n) \leq t_k\} \cap \{u(n) < \mu\}$. We have :

$$\mathbb{P}(u(n) < \mu) \leq \mathbb{P}(\bigcup_{k=1}^{D} A_k) \leq \sum_{k=1}^{D} \mathbb{P}(A_k)$$

Note that by definition of $u(n)$, we have $u(n) < \mu$ if and only if $\hat{\mu}(n) < \mu$ and $N(n)d(\hat{\mu}(n),\mu) > \delta$. Let s be the smallest integer sch that $\delta/(s+1) \leq d(0,\mu)$. If $N(n) \leq s$, then $N(n)d(\hat{\mu},\mu) \leq sd(\hat{\mu},\mu) \leq as\hat{\mu} \underset{\leq}{} \mu sd(0,\mu) by definition of s\delta$. Thus, we can't have $\hat{\mu} < \mu$ and $N(n)d(\hat{\mu},\mu) > \delta$ and $\mathbb{P}(u(n) \underset{<}{} \mu) = 0$. We have for all k such that $t_k \leq s$, $\mathbb{P}(A_k) = 0$ and we have $u(n) > \mu$ when $N(n) \in t_{k-1}+1,\ldots,t_k$ when $t_k \leq s$

Now lets see how $u(n)$ can be upper bounded by μ when $N(n) > s$ For k such that $t_k \geq s$, we note $\tilde{t}_{k-1} = max t_{k-1}, s$ and we take $z < \mu$ such as $d(z,\mu) = \delta/(1+\eta)^k$ and $x \in]0,\mu[$ such that $d(x,\mu) = \delta/N(n)$. We define $\lambda(x) = \log(x(1-\mu)) - log(\mu(1-x)) < 0$

so that we can rewrite $d(x,\mu)$ as$d(x,\mu) = \lambda(x)x - \phi_\mu(\lambda(x))$

— with $N(n) > \tilde{t}_{k-1}$, we have $d(z,\mu) = \frac{\delta}{(1+\mu)^k} \geq \frac{\delta}{(1+\mu)N(n)}$

— with $N(n) \leq t_k$, we have $d(\hat{\mu}(n),\mu) > \frac{\delta}{N(n)} > \frac{\delta}{(1+\eta)^k} = d(z,\mu)$. As $\hat{\mu} < \mu$, we have $\hat{\mu}(n) \leq z$

Hence on the event $\{\tilde{t}_{k-1} < N(n) \leq t_k\} \cap \{\hat{\mu}(n) < \mu\} \cap \{d(\hat{\mu}(n),\mu)$ it holds that $\lambda(z)\hat{\mu}(n) - \phi_\mu(\lambda(z)) \geq \lambda(z)z - \phi_\mu(\lambda(z)) = d(z,\mu) \geq \frac{\delta}{(1+\eta)N(n)}$

It leads to :

$$\begin{aligned}\{\tilde{t}_{k-1} < N(n) \leq t_k\} \cap \{u(n) < \mu\} &\subset \{\lambda(z)\hat{\mu}(n) - \phi_\mu(\lambda(z)) \geq \frac{\delta}{(1+\eta)N(n)}\} \\ &\subset \{\lambda(z)S(n) - N(n)\phi_\mu(\lambda(z)) \geq \frac{\delta}{(1+\eta)}\} \\ &\subset \{W_n^\lambda(z) > \exp\left(\frac{\delta}{(1+\eta)}\right)\}\end{aligned}$$

As $(W_t^\lambda)_{t\geq 0}$ is a supermartingale, $\mathbb{E}[W_n^{\lambda(z)}] \leq \mathbb{E}[W_n^{\lambda(z)}] = 1$, and the Markov inequality yields :

$$\mathbb{P}(\{\tilde{t}_{k-1} < N(n) \leq t_k\} \cap \{u(n) < \mu\}) \leq \mathbb{P}\left(W_n^\lambda(z) > \exp\left(\frac{\delta}{(1+\eta)}\right)\right) \leq \exp\left(-\frac{\delta}{(1+\eta)}\right)$$

As $\eta = 1/(\delta-1)$, $D = \lceil \frac{\log n}{\log 1+\eta} \rceil$ and $\log(1 + 1/(\delta-1)) \geq 1/\delta$, we obtain :

$$\mathbb{P}(u(n) < \mu) \leq \left\lceil \frac{\log n}{\log\left(1 + \frac{1}{\delta-1}\right)} \right\rceil \exp(-\delta + 1) \leq e^{\lceil \delta \log(n) \rceil} \exp(-\delta)$$

□

B.5 Proof of Theorem 3 (Upper-Bound on the Regret of UniRank Assuming a Total Order on Items)

Before proving the regret upper-bound of UniRank, we prove Lemmas 13 and 14 which are respectively bounding the exploration when the leader is the optimal one, and the number of iterations at which the leader is sub-optimal. Finally, the regret upper-bound

of UniRank is given in Appendix B.5.3.

B.5.1 Upper-Bound on the Number of Sub-Optimal Merges of UniRank when the Leader is the Optimal Partition

Lemma 13 (Upper-bound on the number of sub-optimal merges of UniRank when the leader is the optimal partition)**.** *Under the hypotheses of Theorem 3, for any position* $c \in \{2, \dots, L\}$ *UniRank fulfills*

$$\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\left\{\substack{\tilde{\boldsymbol{P}}(t)=\boldsymbol{P}^*,\\ \exists c', P_{c'}(t)=\{\min(c-1,K),c\}}\right\}\right] \leqslant \frac{16}{\tilde{\delta}^*_c \tilde{\Delta}^2_{\min(c-1,K),c}} \log T + \mathcal{O}\left(\log\log T\right).$$

Proof. Let $c \in \{2, \dots, L\}$ be a position, and denote i (respectively j) the item $\min(c-1, K)$ (resp. c). We aim at upper-bounding the number of iterations such that the leader $\tilde{\boldsymbol{P}}(t)$ is the optimal partition $\boldsymbol{P}^*$, and either the subsets $\boldsymbol{P}^*_{c-1} = \{i\}$ and $\boldsymbol{P}^*_c = \{j\}$ are merged in the chosen partition $\boldsymbol{P}(t)$, or $j \in \tilde{\boldsymbol{P}}^*_{K+1}(t)$ is added to the subset $\boldsymbol{P}^*_K = \{i\}$ in the chosen partition $\boldsymbol{P}(t)$. Both situations require $\underline{s}_{i,j}(t)$ to be non-positive.

Let decompose this number of iterations:

$$\begin{aligned}
\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\left\{\substack{\tilde{\boldsymbol{P}}(t)=\boldsymbol{P}^*,\\ \exists c', P_{c'}(t)=\{\min(c-1,K),c\}}\right\}\right] &= \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\left\{\substack{\tilde{\boldsymbol{P}}(t)=\boldsymbol{P}^*,\ \exists c', P_{c'}(t)=\{i,j\},\\ \underline{s}_{i,j}(t)<0}\right\}\right] \\
&\leqslant \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\left\{\substack{\tilde{\boldsymbol{P}}(t)=\boldsymbol{P}^*,\ \exists c', P_{c'}(t)=\{i,j\},\\ \hat{s}_{i,j}(t)<\frac{\tilde{\Delta}_{i,j}}{2}}\right\}\right] \\
&\quad + \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\left\{\substack{\tilde{\boldsymbol{P}}(t)=\boldsymbol{P}^*,\ \exists c', P_{c'}(t)=\{i,j\},\\ T^*_j(t)<\frac{\tilde{\delta}^*_j}{2} t^*_j(t)}\right\}\right] \\
&\quad + \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\left\{\substack{\tilde{\boldsymbol{P}}(t)=\boldsymbol{P}^*,\ \exists c', P_{c'}(t)=\{i,j\},\\ T^*_j(t)\geqslant\frac{\tilde{\delta}^*_j}{2} t^*_j(t),\ \hat{s}_{i,j}(t)\geqslant\frac{\tilde{\Delta}_{i,j}}{2},\\ \underline{s}_{i,j}(t)\leqslant 0}\right\}\right],
\end{aligned}$$

where $t^*_j(t) \overset{def}{=} \sum_{s=1}^{t-1} \mathbb{1}\left\{\tilde{\boldsymbol{P}}(t) = \boldsymbol{P}^*\right\} \mathbb{1}\left\{\exists c, (i,j) \in P_c(s)^2\right\}$,
and $T^*_j(t) \overset{def}{=} \sum_{s=1}^{t-1} \mathbb{1}\left\{\tilde{\boldsymbol{P}}(t) = \boldsymbol{P}^*\right\} \mathbb{1}\left\{\exists c, (i,j) \in P_c(s)^2\right\} \mathbb{1}\{c_i(s) \neq c_j(s)\}$.

Let bound the first term in the right-hand side.

Denote $\Lambda = \left\{t : \tilde{\boldsymbol{P}}(t) = \boldsymbol{P}^*,\ \exists c', P_{c'}(t) = \{i,j\}\right\}$ the set of iterations at which $\tilde{\boldsymbol{P}}(t) = \boldsymbol{P}^*$ and both items i and j are gathered in a subset of $\boldsymbol{P}(t)$. We decompose that set as $\Lambda \subseteq \bigcup_{s\in\mathbb{N}} \Lambda(s)$, with $\Lambda(s) \overset{def}{=} \{t \in \Lambda : t_{i,j}(t) = s\}$. $|\Lambda(s)| \leqslant 1$ as $t_{i,j}(t)$ increases for each $t \in \Lambda$. Note that for each $s \in \mathbb{N}$ and $n \in \Lambda(s)$, $t_{i,j}(n) \geqslant t_{i,j}(n) = s$.

Note also that with the current hypothesis on the order, $i \succ j$, hence by Lemma 10,

$$\mathbb{E}\left[\sum_{t\geq 1} \mathbb{1}\left\{t \in \Lambda, \hat{s}_{i,j}(t) < \frac{\tilde{\Delta}_{i,j}}{2}\right\}\right] = \mathcal{O}(1).$$

The second term may is bounded similarly with the same set Λ but with a different decomposition: $\Lambda \subseteq \bigcup_{s\in\mathbb{N}} \Lambda(s)$, with $\Lambda(s) \overset{def}{=} \{t \in \Lambda : t_j^*(t) = s\}$. $|\Lambda(s)| \leqslant 1$ as $t_j^*(t)$ increases for each $t \in \Lambda$. Note that for each $s \in \mathbb{N}$ and $n \in \Lambda(s)$, $t_j^*(n) \geqslant t_j^*(n) = s$.

Therefore, the same proof as the one used in Lemma 10 gives

$$\mathbb{E}\left[\sum_{t\geq 1} \mathbb{1}\left\{t \in \Lambda, T_j^*(t) < \frac{\tilde{\delta}_j^*}{2} t_j^*(t)\right\}\right] = \mathcal{O}(1)$$

It remains to upper-bound the third term.

Let note

$$C \overset{def}{=} \left\{t \in [T] : \tilde{\boldsymbol{P}}(t) = \boldsymbol{P}^*,\ \exists c', P_{c'}(t) = \{i,j\},\ T_j^*(t) \geqslant \frac{\tilde{\delta}_j^*}{2} t_j^*(t),\ \hat{s}_{i,j}(t) \geqslant \frac{\tilde{\Delta}_{i,j}}{2},\ \underline{s}_{i,j}(t) \leqslant 0\right\}$$

.

Let $t \in C$.

By Pinsker's inequality and as $\underline{s}_{i,j}(t) \leqslant 0$,

$$\begin{aligned}
\frac{1}{2} &\geqslant \frac{\underline{s}_{i,j}(t) + 1}{2} \\
&\geqslant \frac{\hat{s}_{i,j}(t) + 1}{2} - \sqrt{\frac{\log(\tilde{t}_{\boldsymbol{P}^*}(t)) + 3\log(\log(\tilde{t}_{\boldsymbol{P}^*}(t)))}{2T_{i,j}(t)}} \\
&\geqslant \frac{\tilde{\Delta}_{i,j}}{4} + \frac{1}{2} - \sqrt{\frac{\log(\tilde{t}_{\boldsymbol{P}^*}(t))) + 3\log(\log(\tilde{t}_{\boldsymbol{P}^*}(t))))}{2T_{i,j}(t)}}.
\end{aligned}$$

Hence, $T_{i,j}(t) \leqslant \frac{8\log(\tilde{t}_{\boldsymbol{P}^*}(t))) + 24\log(\log(\tilde{t}_{\boldsymbol{P}^*}(t))))}{\tilde{\Delta}_{i,j}^2}$ as $\tilde{\Delta}_{i,j} > 0$ given Lemma 7. Then, by definition of C and as (i) $\tilde{t}_{\boldsymbol{P}^*}(t) \leqslant t \leqslant T$, (ii) $T_j^*(t) \leqslant T_{i,j}(t)$, and (iii) $\tilde{\delta}_j^* \geqslant \tilde{\delta}_{i,j} > 0$ given Lemma 7, $t_j^*(t) \leqslant \frac{2T_j^*(t)}{\tilde{\delta}_j^*} \leqslant \frac{2T_{i,j}(t)}{\tilde{\delta}_j^*} \leqslant \frac{16\log(T) + 48\log(\log(T))}{\tilde{\delta}_j^* \tilde{\Delta}_{i,j}^2}$.

Therefore, $C \subseteq \left\{t \in [T] : \tilde{\boldsymbol{P}}(t) = \boldsymbol{P}^*,\ \exists c', P_{c'}(t) = \{i,j\},\ t_j^*(t) \leqslant \frac{16\log(T) + 48\log(\log(T))}{\tilde{\delta}_j^* \tilde{\Delta}_{i,j}^2}\right\}$,

and

$$\mathbb{E}\left[\sum_{t=1}^{T}\mathbb{1}\left\{\begin{array}{c}\tilde{\boldsymbol{P}}(t)=\boldsymbol{P}^*,\ \exists c',P_{c'}(t)=\{i,j\},\\ T_j^*(t)\geqslant\frac{\tilde{\delta}_j^*}{2}t_j^*(t),\ \hat{s}_{i,j}(t)\geqslant\frac{\tilde{\Delta}_{i,j}}{2},\\ \underline{s}_{i,j}(t)\leqslant 0\end{array}\right\}\right] = \mathbb{E}\left[|C|\right]$$
$$\leqslant \mathbb{E}\left[\left|\left\{\begin{array}{c}t\in[T]:\ \tilde{\boldsymbol{P}}(t)=\boldsymbol{P}^*,\ \exists c',P_{c'}(t)=\{i,j\},\\ t_j^*(t)\leqslant\frac{16\log(T)+48\log(\log(T))}{\tilde{\delta}_j^*\tilde{\Delta}_{i,j}^2}\end{array}\right\}\right|\right]$$
$$\leqslant \frac{16\log(T)+48\log(\log(T))}{\tilde{\delta}_j^*\tilde{\Delta}_{i,j}^2},$$

which concludes the proof.

□

B.5.2 Upper-Bound on the Expected Number of Iterations at which the Leader is not the Optimal Partition

Lemma 14 (Upper-bound on the expected number of iterations at which the leader is not the optimal partition)**.** *Under the hypotheses of Theorem 3, UniRank fulfills*

$$\mathbb{E}\left[\sum_{t=1}^{T}\mathbb{1}\{\tilde{\boldsymbol{P}}(t)\neq\boldsymbol{P}^*\}\right] = \mathcal{O}\left(\log\log T\right).$$

Proof. Let $\tilde{\boldsymbol{P}}\neq\boldsymbol{P}^*$ be an ordered partition of items of size d, and let upper-bound the expected number of iterations at which $\tilde{\boldsymbol{P}}(t)=\tilde{\boldsymbol{P}}$ by $\mathcal{O}\left(\log\log T\right)$ As there is a finite number of partitions, this will conclude the proof.

In this proof, for any couple of items (i,j) we denote $\tilde{t}_{i,j}(s)\overset{def}{=}\sum_{s=1}^{t-1}\mathbb{1}\left\{\tilde{\boldsymbol{P}}(t)=\tilde{\boldsymbol{P}},\exists c,(i,j)\in P_c(s)^2\right.$ the number of iteration at which both items have been gathered in the same subset of $\boldsymbol{P}(s)$ while the leader was $\tilde{\boldsymbol{P}}$.

The proof depends on the difference between $\tilde{\boldsymbol{P}}$ and $\boldsymbol{P}^*$. By Lemma 5,

— either there exists $c\in[\tilde{d}-1]$ such that $|\tilde{P}_c|>1$;
— there exists $c\in[\tilde{d}-1]\setminus\{1\}$ such that $\tilde{P}_{c-1}=\{i'\}$, $\tilde{P}_c=\{i\}$, $i'\succ i$, and there exists $j\in\tilde{P}_{c+1}$ such that $j\succ i$;
— or for $c=1$, $\tilde{P}_c=\{i\}$ and there exists $j\in\tilde{P}_{c+1}$ such that $j\succ i$;

We first upper-bound the expected number of iterations at which $\tilde{\boldsymbol{P}}(t)=\tilde{\boldsymbol{P}}$ under the first condition, and then prove a similar upper-bound under both other conditions.

Assume that there exists $c \in [\tilde{d}-1]$ such that $|\tilde{P}_c| > 1$ Let t be an iteration such that $\tilde{\boldsymbol{P}}(t) = \tilde{\boldsymbol{P}}$. As $\succ$ is a strict total order, there exists an item $i^* \in \tilde{P}_c$ which is strictly greater than other items. Moreover, by Assumption 5 and by design of the algorithm, if for each couple of items $(i,j) \in \tilde{P}_c^2$ such that $i \neq j$, the sign of $\tilde{s}_{i,j}(t) \overset{def}{=} \hat{s}_{i,j}(t) - \sqrt{\frac{\log\log t}{T_{i,j}(t)}}$ would be the same as the sign of $\tilde{\Delta}_{i,j}$, then i^* would be alone in $\tilde{P}_c(t)$. So these signs disagree for at least one of these couples of items. Let control the number of iteration at which this is true by considering the following decomposition:

$$\left\{t : \tilde{\boldsymbol{P}}(t) = \tilde{\boldsymbol{P}}\right\} \subseteq \bigcup_{(i,j)\in\tilde{P}_c^2 : i\neq j} A_{i,j} \cup B_{i,j} \cup C_{i,j},$$

where

$$A_{i,j} \overset{def}{=} \left\{t : \tilde{\boldsymbol{P}}(t) = \tilde{\boldsymbol{P}}, T_{i,j}(t) < \frac{\tilde{\delta}_{i,j}}{2} t_{i,j}(t)\right\},$$

$$B_{i,j} \overset{def}{=} \left\{t : \tilde{\boldsymbol{P}}(t) = \tilde{\boldsymbol{P}}, \frac{\hat{s}_{i,j}(t)}{\tilde{\Delta}_{i,j}} < \frac{1}{2}\right\},$$

and

$$C_{i,j} \overset{def}{=} \left\{t : \tilde{\boldsymbol{P}}(t) = \tilde{\boldsymbol{P}}, T_{i,j}(t) \geqslant \frac{\tilde{\delta}_{i,j}}{2} t_{i,j}(t), \frac{\hat{s}_{i,j}(t)}{\tilde{\Delta}_{i,j}} \geqslant \frac{1}{2}, \tilde{s}_{i,j}(t)\tilde{\Delta}_{i,j} \leqslant 0, \right\}.$$

Let $(i,j) \in \tilde{P}_c^2$ be a couple of items such that $i \neq j$, and let first upper-bound the expected size of $A_{i,j}$ and $B_{i,j}$, and then the expected size of $C_{i,j}$.

Note that at each iteration such that $\tilde{\boldsymbol{P}}(t) = \tilde{\boldsymbol{P}}$, i and j are in the same subset of the partition $\boldsymbol{P}(t)$, therefore $\tilde{t}_{i,j}(t) = \tilde{t}_{\tilde{\boldsymbol{P}}}(t)$.

Denote $\Lambda = \left\{t : \tilde{\boldsymbol{P}}(t) = \tilde{\boldsymbol{P}}\right\}$ the set of iterations at which $\tilde{\boldsymbol{P}}(t) = \tilde{\boldsymbol{P}}$, and decompose that set as $\Lambda \subseteq \bigcup_{s\in\mathbb{N}} \Lambda(s)$, with $\Lambda(s) \overset{def}{=} \{t \in \Lambda : \tilde{t}_{\tilde{\boldsymbol{P}}}(t) = s\}$. $|\Lambda(s)| \leqslant 1$ as $\tilde{t}_{\tilde{\boldsymbol{P}}}(t)$ increases for each $t \in \Lambda$. Note that for each $s \in \mathbb{N}$ and $n \in \Lambda(s)$, $t_{i,j}(n) \geqslant \tilde{t}_{i,j}(n) = \tilde{t}_{\tilde{\boldsymbol{P}}}(t) = s$.

Then by Lemma 10

$$\mathbb{E}\left[|A_{i,j}|\right] = \mathbb{E}\left[\sum_{t\geq 1} \mathbb{1}\left\{t \in \Lambda, T_{i,j}(t) < \frac{\tilde{\delta}_{i,j}}{2} t_{i,j}(t)\right\}\right] = \mathcal{O}(1)$$

and

$$\mathbb{E}\left[|B_{i,j}|\right] = \mathbb{E}\left[\sum_{t\geq 1} \mathbb{1}\left\{t \in \Lambda, \frac{\hat{s}_{i,j}(t)}{\tilde{\Delta}_{i,j}} < \frac{1}{2}\right\}\right] = \mathcal{O}(1).$$

Let now upper-bound the expected size of $C_{i,j}$ by considering separately two situations for items i and j: (1) $i \succ j$, and (2) $j \succ i$.

Situation (1): $i \succ j$ Then $\tilde{\Delta}_{i,j} > 0$.

Let $t \in C_{i,j}$. As $\tilde{s}_{i,j}(t) \leqslant 0$, $t \leqslant T$, and $\tilde{t}_{\tilde{\boldsymbol{P}}}(t) = \tilde{t}_{i,j}(t) \leqslant t_{i,j}(t) \leqslant \frac{2}{\tilde{\delta}_{i,j}} T_{i,j}(t)$,

$$\begin{aligned} 0 &\geqslant \tilde{s}_{i,j}(t) \\ &= \hat{s}_{i,j}(t) - \sqrt{\frac{\log\log t}{T_{i,j}(t)}} \\ &\geqslant \frac{\tilde{\Delta_{i,j}}}{2} - \sqrt{\frac{2\log\log T}{\tilde{\delta}_{i,j}\tilde{t}_{\tilde{\boldsymbol{P}}}(t)}}. \end{aligned}$$

Hence, $\tilde{t}_{\tilde{\boldsymbol{P}}}(t) \leqslant \frac{8\log\log T}{\tilde{\delta}_{i,j}\tilde{\Delta}_{i,j}^2}$, and

$$\begin{aligned} \mathbb{E}\left[|C_{i,j}|\right] &\leqslant \mathbb{E}\left[\sum_{t\geq 1} \mathbb{1}\left\{t \in C_{i,j} : \tilde{t}_{\tilde{\boldsymbol{P}}}(t) \leqslant \frac{8\log\log T}{\tilde{\delta}_{i,j}\tilde{\Delta}_{i,j}^2}\right\}\right] \\ &= \mathcal{O}\left(\log\log T\right) \end{aligned}$$

Situation (2): $j \succ i$ Then $\tilde{\Delta}_{i,j} < 0$.

Let $t \in C_{i,j}$. As $\tilde{s}_{i,j}(t) \geqslant 0$,

$$\begin{aligned} 0 &\leqslant \tilde{s}_{i,j}(t) \\ &= \hat{s}_{i,j}(t) - \sqrt{\frac{\log\log t}{T_{i,j}(t)}} \\ &\leqslant \frac{\tilde{\Delta}_{i,j}}{2} - \sqrt{\frac{\log\log t}{T_{i,j}(t)}} \\ &< 0. \end{aligned}$$

Which is absurd. Hence, $C_{i,j} = \varnothing$, and $\mathbb{E}\left[|C_{i,j}|\right] = 0$

Overall, if there exists $c \in [\tilde{d} - 1]$ such that $|\tilde{P}_c| > 1$,

$$\begin{aligned} \mathbb{E}\left[\mathbb{1}\{\tilde{\boldsymbol{P}}(t) = \tilde{\boldsymbol{P}}\}\right] &\leqslant \sum_{(i,j)\in\tilde{P}_c^2 : i\neq j} \mathbb{E}\left[|A_{i,j}|\right] + \mathbb{E}\left[|B_{i,j}|\right] + \mathbb{E}\left[|C_{i,j}|\right] \\ &= \mathcal{O}(1) + \mathcal{O}(1) + (\mathcal{O}(\log\log T) + 0) \\ &= \mathcal{O}(\log\log T) \end{aligned}$$

Assume that there exists $c \in [\tilde{d}-1]$ such that $\tilde{P}_c = \{i\}$ and there exists $j \in \tilde{P}_{c+1}$ such that $j \succ i$ Let's now consider the two last possible conditions w.r.t. $\tilde{\boldsymbol{P}}$. Under these conditions, there may exist an item i' or not. To shorten the proof, we use notations including i' even when it does not exist. Corresponding sets (respectively counts) have to be read as empty (resp. equal to 0) when i' does not exist.

Also remark that under this condition, $\tilde{d} = K+1$, for each index $c' \in [K]$, $|P_{c'}| = 1$, and, in P_{c+1}, j is the unique item more attractive than item i.

To bound $\mathbb{E}\left[\mathbb{1}\{\tilde{\boldsymbol{P}}(t) = \tilde{\boldsymbol{P}}\}\right]$, we use the decomposition $\{t \in [T] : \tilde{\boldsymbol{P}}(t) = \tilde{\boldsymbol{P}}\} = A \cup B$ where

$$A = \left\{t : \tilde{\boldsymbol{P}}(t) = \tilde{\boldsymbol{P}}, \tilde{t}_{i,j}(t) \geqslant \frac{1}{2}\tilde{t}_{\tilde{\boldsymbol{P}}}(t)\right\}$$

and

$$B = \left\{t : \tilde{\boldsymbol{P}}(t) = \tilde{\boldsymbol{P}}, \forall j \in \tilde{P}^+_{c+1}, \tilde{t}_{i,j}(t) < \frac{1}{2}\tilde{t}_{\tilde{\boldsymbol{P}}}(t)\right\}.$$

Hence,

$$\mathbb{E}\left[\mathbb{1}\{\tilde{\boldsymbol{P}}(t) = \tilde{\boldsymbol{P}}\}\right] \leqslant \sum_{j \in \tilde{P}^+_{c+1}} \mathbb{E}\left[|A|\right] + \mathbb{E}\left[|B|\right].$$

Bound on $\mathbb{E}\left[|A|\right]$ Let $j \in \tilde{P}^+_{c+1}$ be an item. By design of UniRank, $\tilde{s}_{j,i}(t) \leqslant 0$ as i is in a subset before the one of j in $\tilde{\boldsymbol{P}}(t)$.

Let's decompose A as $A \subseteq \bigcup_{s \in \mathbb{N}} \Lambda(s)$, with $\Lambda(s) \overset{def}{=} \{t \in A : \tilde{t}_{\tilde{\boldsymbol{P}}}(t) = s\}$. $|\Lambda(s)| \leqslant 1$ as $\tilde{t}_{\tilde{\boldsymbol{P}}}(t)$ increases for each $t \in A$. Note that for each $s \in \mathbb{N}$ and $n \in \Lambda(s)$, $t_{i,j}(n) \geqslant \tilde{t}_{i,j}(n) \geqslant \frac{1}{2}\tilde{t}_{\tilde{\boldsymbol{P}}}(t) = \frac{1}{2}s$.

As $j \succ i$, by Lemma 10

$$\mathbb{E}\left[\sum_{t \geq 1} \mathbb{1}\left\{t \in A, T_{i,j}(t) < \frac{\tilde{\delta}_{i,j}}{2} t_{i,j}(t)\right\}\right] = \mathcal{O}(1)$$

and

$$\mathbb{E}\left[\sum_{t \geq 1} \mathbb{1}\left\{t \in A, \hat{s}_{j,i}(t) < \frac{\tilde{\Delta}_{j,i}}{2}\right\}\right] = \mathcal{O}(1).$$

Let $t \in A$ such that $T_{i,j}(t) \geqslant \frac{\tilde{\delta}_{i,j}}{2} t_{i,j}(t)$ and $\hat{s}_{j,i}(t) \geqslant \frac{\tilde{\Delta}_{j,i}}{2}$.

As $\tilde{s}_{j,i}(t) \leqslant 0$,

$$\begin{aligned} 0 &\geqslant \tilde{s}_{j,i}(t) \\ &= \hat{s}_{j,i}(t) - \sqrt{\frac{\log\log t}{T_{i,j}(t)}} \\ &\leqslant \frac{\tilde{\Delta}_{j,i}}{2} - \sqrt{\frac{\log\log t}{T_{i,j}(t)}}. \end{aligned}$$

Hence, $T_{ij}(t) \leqslant \frac{4}{\tilde{\Delta}_{j,i}^2} \log\log t \leqslant \frac{4}{\tilde{\Delta}_{j,i}^2} \log\log T$.

Remind that by definition of A,

$$\tilde{t}_{\tilde{\boldsymbol{P}}(t)}(t) \leqslant 2\tilde{t}_{i,j}(t) \leqslant 2t_{i,j}(t) \leqslant \frac{4}{\tilde{\delta}_{i,j}} T_{i,j}(t) \leqslant \frac{16}{\tilde{\delta}_{i,j}\tilde{\Delta}_{j,i}} \log\log T.$$

Therefore, $\left\{t \in A:\ T_{i,j}(t) \geqslant \frac{\tilde{\delta}_{i,j}}{2} t_{i,j}(t),\ \hat{s}_{i,j}(t) \leqslant \frac{\tilde{\Delta}_{i,j}}{2}\right\} \subseteq \left\{t \in A:\ \tilde{t}_{\tilde{\boldsymbol{P}}(t)}(t) \leqslant \frac{16}{\tilde{\delta}_{i,j}\tilde{\Delta}_{j,i}} \log\log T\right\}$, and

$$\begin{aligned} E\left[|A|\right] &\leqslant \mathbb{E}\left[\sum_{t\geq 1} \mathbb{1}\left\{t \in A : T_{i,j}(t) < \frac{\tilde{\delta}_{i,j}}{2} t_{i,j}(t)\right\}\right] \\ &\quad + \mathbb{E}\left[\sum_{t\geq 1} \mathbb{1}\left\{t \in A : \hat{s}_{i,j}(t) > \frac{\tilde{\Delta}_{i,j}}{2}\right\}\right] \\ &\quad + \mathbb{E}\left[\sum_{t\geq 1} \mathbb{1}\left\{t \in A : T_{i,j}(t) \geqslant \frac{\tilde{\delta}_{i,j}}{2} t_{i,j}(t), \hat{s}_{i,j}(t) \leqslant \frac{\tilde{\Delta}_{i,j}}{2}\right\}\right] \\ &\leqslant \mathbb{E}\left[\sum_{t\geq 1} \mathbb{1}\left\{t \in A : T_{i,j}(t) < \frac{\tilde{\delta}_{i,j}}{2} t_{i,j}(t)\right\}\right] \\ &\quad + \mathbb{E}\left[\sum_{t\geq 1} \mathbb{1}\left\{t \in A : \hat{s}_{i,j}(t) > \frac{\tilde{\Delta}_{i,j}}{2}\right\}\right] \\ &\quad + \mathbb{E}\left[\sum_{t\geq 1} \mathbb{1}\left\{t \in A : \tilde{t}_{\tilde{\boldsymbol{P}}(t)}(t) \leqslant \frac{16}{\tilde{\delta}_{i,j}\tilde{\Delta}_{j,i}} \log\log T\right\}\right] \\ &= \mathcal{O}\left(1\right) + \mathcal{O}\left(1\right) + \mathcal{O}\left(\log\log T\right) \\ &= \mathcal{O}\left(\log\log T\right). \end{aligned}$$

Bound on $\mathbb{E}\left[|B|\right]$ We first split B in two parts: $B = B^{t_0} \cup B^T_{t_0}$, where $B^{t_0} \stackrel{def}{=} \{t \in B : \tilde{t}_{\tilde{\boldsymbol{P}}}(t) \leqslant t_0\}$, $B^T_{t_0} \stackrel{def}{=} \{t \in B : \tilde{t}_{\tilde{\boldsymbol{P}}}(t) > t_0\}$, and t_0 is chosen as small as possible to satisfy

three constraints required in the rest of the proof. Namely,

$$t_0 = \max\left\{20, \inf\left\{t : \sqrt{\frac{5\log(t)+15\log(\log(t))}{\tilde{\delta}_{i',i}t}} < \frac{\tilde{\Delta}_{i',i}}{8}\right\},\right.$$
$$\left.\max_{k\in P_{c+1}\setminus\{j\}} \inf\left\{t : \sqrt{\frac{5\log(t)+15\log(\log(t))}{\tilde{\delta}_{k,i}t}} < \frac{\tilde{\Delta}_{i,k}}{8}\right\}\right\}.$$

Note that t_0 only depends on $\tilde{\delta}_{i',i}$, $\tilde{\Delta}_{i',i}$ and $\tilde{\delta}_{k,i}$, $\tilde{\Delta}_{i,k}$ for $k \in P_{c+1} \setminus \{j\}$.

We also define

— $D \overset{def}{=} \left\{t \in [T] : \tilde{\boldsymbol{P}}(t) = \tilde{\boldsymbol{P}}, c > 1, \exists c', P_{c'}(t) = \{i', i\}, T_{i',i}(t) < \frac{\tilde{\delta}_{i',i}}{2} t_{i',i}(t)\right\}$,

— $D'_k \overset{def}{=} \left\{t \in [T] : \tilde{\boldsymbol{P}}(t) = \tilde{\boldsymbol{P}}, c > 1, \exists c', P_{c'}(t) = \{k, i\}, T_{k,i}(t) < \frac{\tilde{\delta}_{k,i}}{2} t_{k,i}(t)\right\}$, for each $k \in P_{c+1} \setminus \{j\}$

— $E \overset{def}{=} \left\{t \in [T] : \tilde{\boldsymbol{P}}(t) = \tilde{\boldsymbol{P}}, c > 1, \exists c', P_{c'}(t) = \{i', i\}, \hat{s}_{i',i}(t) < \frac{\tilde{\Delta}_{i',i}}{2}\right\}$,

— $E'_k \overset{def}{=} \left\{t \in [T] : \tilde{\boldsymbol{P}}(t) = \tilde{\boldsymbol{P}}, c > 1, \exists c', P_{c'}(t) = \{k, i\}, \hat{s}_{i,k}(t) < \frac{\tilde{\Delta}_{i,k}}{2}\right\}$, for each $k \in P_{c+1} \setminus \{j\}$

— $F \overset{def}{=} \{t \in [T] : \tilde{\boldsymbol{P}}(t) = \tilde{\boldsymbol{P}}, \frac{s_{i,j}(t)+1}{2} \geqslant \frac{\tilde{\Delta}_{i,j}+1}{2}\}$.

Finally, for any $t \in \mathbb{N}$ we denote $a(t)$ the event $\{c > 1, \exists c', P_{c'}(t) = \{i', i\}\}$, $a'_k(t)$ the event $\{\exists c', P_{c'}(t) = \{k, i\}\}$ for each $k \in \tilde{P}_{c+1} \setminus \{j\}$, and $b(t)$ the event $\{\forall c', P_{c'}(t) \neq \{i', i\}, \forall k \in \tilde{P}_{c+1} \forall c', P_{c'}(t) \neq \{k, i\}\}$, which we associate to their respective number of occurrences while the leader is $\tilde{\boldsymbol{P}}$:

— $\tilde{t}_a(t) \overset{def}{=} \sum_{s=1}^{t-1} \mathbb{1}\left\{\tilde{\boldsymbol{P}}(s) = \tilde{\boldsymbol{P}}, c > 1, \exists c', P_{c'}(t) = \{i', i\}\right\} = \tilde{t}_{i',i}(t)$,

— $\tilde{t}_{a'_k}(t) \overset{def}{=} \sum_{s=1}^{t-1} \mathbb{1}\left\{\tilde{\boldsymbol{P}}(s) = \tilde{\boldsymbol{P}}, \exists c', P_{c'}(t) = \{k, i\}\right\} = \tilde{t}_{k,i}(t)$,

— $\tilde{t}_b(t) \overset{def}{=} \sum_{s=1}^{t-1} \mathbb{1}\left\{\tilde{\boldsymbol{P}}(s) = \tilde{\boldsymbol{P}}, \forall c', P_{c'}(t) \neq \{i', i\}, \forall c', P_{c'}(t) \neq \{i, j\}\right\}$.

Let $t \in B^T_{t_0}$. By design of UniRank,

$$\tilde{t}_{\tilde{\boldsymbol{P}}}(t) = \tilde{t}_{i,j}(t) + \tilde{t}_a(t) + \sum_{k\in\tilde{P}_{c+1}\setminus\{j\}} \tilde{t}_{a'_k}(t) + \tilde{t}_b(t).$$

Therefore, as $\tilde{t}_{i,j}(t) < \frac{1}{2}\tilde{t}_{\tilde{\boldsymbol{P}}}(t)$ and by definition of t_0, either $\tilde{t}_a(t) \geqslant \frac{1}{\gamma}\tilde{t}_{\tilde{\boldsymbol{P}}}(t) + 1$, or there exist $k \in P_{c+1} \setminus \{j\}$ such that $\tilde{t}_{a'_k}(t) \geqslant \frac{1}{\gamma}\tilde{t}_{\tilde{\boldsymbol{P}}}(t) + 1$, or $\tilde{t}_b(t) \geqslant \frac{1}{\gamma}\tilde{t}_{\tilde{\boldsymbol{P}}}(t) + 1$. Let denote e the index of the event such that $\tilde{t}_e(t) \geqslant \frac{1}{\gamma}\tilde{t}_{\tilde{\boldsymbol{P}}}(t) + 1$ and elicit an iteration $\psi(t)$ with specific properties.

We denote s' the first iteration such that $\tilde{t}_e(t) \geqslant \frac{1}{\gamma}\tilde{t}_{\tilde{\boldsymbol{P}}}(t) + 1$. At this iteration, $\tilde{t}_e(s') =$

$\tilde{t}_e(s'-1)+1$, meaning that $\tilde{\boldsymbol{P}}(s'-1)=\tilde{\boldsymbol{P}}$, $e(s'-1)$ is true, and $\tilde{t}_e(s'-1)\geqslant\frac{1}{\gamma}\tilde{t}_{\tilde{\boldsymbol{P}}}(t)$. Therefore, the set $\{s\in[t]:\tilde{\boldsymbol{P}}(s)=\tilde{\boldsymbol{P}},e(t),\tilde{t}_e(s)\geqslant\frac{1}{\gamma}\tilde{t}_{\tilde{\boldsymbol{P}}}(t)\}$ is non-empty. We define $\psi(t)$ as the minimum on this set

$$\psi(t)\stackrel{def}{=}\min\left\{s\in[t]:\tilde{\boldsymbol{P}}(s)=\tilde{\boldsymbol{P}},e(t),\tilde{t}_e(s)\geqslant\frac{1}{\gamma}\tilde{t}_{\tilde{\boldsymbol{P}}}(t)\right\}.$$

Let prove by contradiction that $\psi(t)\in D\cup E\cup F\cup\bigcup_{k\in P_{c+1}\setminus\{j\}}(D'_k\cup E'_k)$. Assume that $\psi(t)\notin D\cup E\cup F\cup\bigcup_{k\in P_{c+1}\setminus\{j\}}(D'_k\cup E'_k)$. Either e is a, or e is a'_k for some item k, or e is b.

If e is a, meaning $a(\psi(t))$ is true Then by design of UniRank, $\underline{s}_{i',i}(\psi(t))\leqslant 0$. Moreover, since $\tilde{\boldsymbol{P}}(\psi(t))=\tilde{\boldsymbol{P}}$ and there exists c' such that $P_{c'}(\psi(t))=\{i',i\}$, and $\psi(t)\notin D\cup E$, $T_{i',i}(\psi(t))\geqslant\frac{\tilde{\delta}_{i',i}}{2}t_{i',i}(\psi(t))$ and $\hat{s}_{i',i}(\psi(t))\geqslant\frac{\tilde{\Delta}_{i',i}}{2}$.

Therefore,

$$T_{i',i}(\psi(t))\geqslant\frac{\tilde{\delta}_{i',i}}{2}t_{i',i}(\psi(t))\geqslant\frac{\tilde{\delta}_{i',i}}{2}\tilde{t}_a(\psi(t))\geqslant\frac{\tilde{\delta}_{i',i}}{2}\frac{1}{\gamma}\tilde{t}_{\tilde{\boldsymbol{P}}}(t)$$

and by Pinsker's inequality and the fact that $\psi(t)\leqslant t$ and $\tilde{t}_{\tilde{\boldsymbol{P}}}(s)$ is non-decreasing in s, and $\tilde{t}_{\tilde{\boldsymbol{P}}}(t)>t_0$,

$$\begin{aligned}\frac{1}{2}\geqslant\frac{\underline{s}_{i',i}(\psi(t))+1}{2}&\geqslant\frac{\hat{s}_{i',i}(\psi(t))+1}{2}-\sqrt{\frac{\log(\tilde{t}_{\tilde{\boldsymbol{P}}}(\psi(t)))+3\log(\log(\tilde{t}_{\tilde{\boldsymbol{P}}}(\psi(t))))}{2T_{i',i}(\psi(t))}}\\&\geqslant\frac{1}{2}+\frac{\tilde{\Delta}_{i',i}}{4}-\sqrt{\frac{5\log(\tilde{t}_{\tilde{\boldsymbol{P}}}(t))+15\log(\log(\tilde{t}_{\tilde{\boldsymbol{P}}}(t)))}{\tilde{\delta}_{i',i}\tilde{t}_{\tilde{\boldsymbol{P}}}(t)}}\\&\geqslant\frac{1}{2}+\frac{\tilde{\Delta}_{i',i}}{4}-\frac{\tilde{\Delta}_{i',i}}{8}\\&=\frac{1}{2}+\frac{\tilde{\Delta}_{i',i}}{8}\end{aligned}$$

which contradicts the fact that $\tilde{\Delta}_{i',i}>0$.

If e is a'_k for some $k\in P_{c+1}\setminus\{j\}$, meaning $a'_k(\psi(t))$ is true Then by design of UniRank, $\underline{s}_{i,k}(\psi(t))\leqslant 0$. Moreover, since $\tilde{\boldsymbol{P}}(\psi(t))=\tilde{\boldsymbol{P}}$ and there exists c' such that $P_{c'}(\psi(t))=\{k,i\}$, and $\psi(t)\notin D'_k\cup E'_k$, $T_{k,i}(\psi(t))\geqslant\frac{\tilde{\delta}_{k,i}}{2}t_{k,i}(\psi(t))$ and $\hat{s}_{i,k}(\psi(t))\geqslant\frac{\tilde{\Delta}_{i,k}}{2}$.

Therefore,

$$T_{k,i}(\psi(t)) \geqslant \frac{\tilde{\delta}_{k,i}}{2} t_{k,i}(\psi(t)) \geqslant \frac{\tilde{\delta}_{k,i}}{2} \tilde{t}_a(\psi(t)) \geqslant \frac{\tilde{\delta}_{k,i}}{2} \frac{1}{\gamma} \tilde{t}_{\tilde{\boldsymbol{P}}}(t)$$

and by Pinsker's inequality and the fact that $\psi(t) \leqslant t$ and $\tilde{t}_{\tilde{\boldsymbol{P}}}(s)$ is non-decreasing in s, and $\tilde{t}_{\tilde{\boldsymbol{P}}}(t) > t_0$,

$$\begin{aligned}
\frac{1}{2} \geqslant \frac{\underline{s}_{i,k}(\psi(t)) + 1}{2} &\geqslant \frac{\hat{s}_{i,k}(\psi(t)) + 1}{2} - \sqrt{\frac{\log(\tilde{t}_{\tilde{\boldsymbol{P}}}(\psi(t))) + 3\log(\log(\tilde{t}_{\tilde{\boldsymbol{P}}}(\psi(t))))}{2T_{k,i}(\psi(t))}} \\
&\geqslant \frac{1}{2} + \frac{\tilde{\Delta}_{i,k}}{4} - \sqrt{\frac{5\log(\tilde{t}_{\tilde{\boldsymbol{P}}}(t)) + 15\log(\log(\tilde{t}_{\tilde{\boldsymbol{P}}}(t)))}{\tilde{\delta}_{k,i}\tilde{t}_{\tilde{\boldsymbol{P}}}(t)}} \\
&\geqslant \frac{1}{2} + \frac{\tilde{\Delta}_{i,k}}{4} - \frac{\tilde{\Delta}_{i,k}}{8} \\
&= \frac{1}{2} + \frac{\tilde{\Delta}_{i,k}}{8}
\end{aligned}$$

which contradicts the fact that $\tilde{\Delta}_{i,k} > 0$.

If e is b, meaning $b(\psi(t))$ is true Then by design of UniRank, $\underline{s}_{i,j}(\psi(t)) > 0$. Moreover, since $\tilde{\boldsymbol{P}}(\psi(t)) = \tilde{\boldsymbol{P}}$ and $\psi(t) \notin F$, $\frac{\underline{s}_{i,j}(\psi(t))+1}{2} < \frac{\tilde{\Delta}_{i,j}+1}{2}$.

Therefore, $\frac{1}{2} < \frac{\underline{s}_{i,j}(\psi(t))+1}{2} < \frac{\tilde{\Delta}_{i,j}+1}{2}$ which contradicts the fact that $\tilde{\Delta}_{i,j} < 0$.

Overall, $\psi(t) \notin D \cup E \cup F \cup \bigcup_{k \in P_{c+1} \setminus \{j\}} (D'_k \cup E'_k)$ leads to a contradiction while either e is a, or e is a'_k, or e is b. So, for any $t \in B^T_{t_0}$, $\psi(t) \in D \cup E \cup F \cup \bigcup_{k \in P_{c+1} \setminus \{j\}} (D'_k \cup E'_k)$, and $B^T_{t_0} \subseteq \bigcup_{n \in D \cup E \cup F \cup \bigcup_{k \in P_{c+1} \setminus \{j\}} (D'_k \cup E'_k)} B^T_{t_0} \cap \{t \in [T] : \psi(t) = n\}$. Let n be in $D \cup E \cup F \cup \bigcup_{k \in P_{c+1} \setminus \{j\}} (D'_k \cup E'_k)$. For any t in $B^T_{t_0} \cap \{t \in [T] : \psi(t) = n\}$, there exists an index $e \in \{a, a'_k, b\}$ such that $\tilde{t}_e(n) = \lceil \frac{1}{\gamma} \tilde{t}_{\tilde{\boldsymbol{P}}}(t) \rceil$, and $\tilde{t}_e(n+1) = \tilde{t}_e(n) + 1$. So $|B^T_{t_0} \cap \{t \in [T] : \psi(t) = n\}| \leqslant \gamma$ and

$$\mathbb{E}[|B|] \leqslant t_0 + \mathbb{E}\left[|B^T_{t_0}|\right] \leqslant t_0 + \gamma(\mathbb{E}[|D|] + \mathbb{E}[|E|] + \mathbb{E}[|F|]).$$

It remains to upper-bound $\mathbb{E}[|D|]$, $\mathbb{E}[|E|]$, and $\mathbb{E}[|F|]$ to conclude the proof.

Bound on $\mathbb{E}[|D|]$ and $\mathbb{E}[|E|]$ The upper-bound on $\mathbb{E}[|D|]$ and $\mathbb{E}[|E|]$ is obtained through Lemma 10. Let $\Lambda \stackrel{def}{=} \left\{t \in [T] : \tilde{\boldsymbol{P}}(t) = \tilde{\boldsymbol{P}}, c > 1, \exists c', P_{c'}(t) = \{i', i\}\right\}$ and uses the decomposition $\Lambda \subseteq \bigcup_{s \in \mathbb{N}} \Lambda(s)$, where $\Lambda(s) \stackrel{def}{=} \{t \in \Lambda : t_{i',i}(t) = s\}$. $|\Lambda(s)| \leqslant 1$ as $t_{i',i}(t)$

increases for each $t \in \Lambda$. Note that for each $s \in \mathbb{N}$ and $n \in \Lambda(s)$, $t_{i',i}(n) \geqslant t_{i',i}(n) = s$. Then, by Lemma 10, as $i' \succ i$

$$\mathbb{E}\left[|D|\right] = \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{t \in \Lambda : T_{i',i}(t) < \frac{\tilde{\delta}_{i',i}}{2} t_{i',i}(t)\}\right] = \mathcal{O}(1)$$

and

$$\mathbb{E}\left[|D|\right] = \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{t \in \Lambda : \hat{s}_{i',i}(t) < \frac{\tilde{\Delta}_{i',i}}{2}\}\right] = \mathcal{O}(1).$$

Bound on $\mathbb{E}\left[|F|\right]$ By Lemma 12, $\mathbb{E}\left[|F|\right] = O(\log(\log(T)))$.

Overall $\mathbb{E}\left[\mathbb{1}\{\tilde{\boldsymbol{P}}(t) = \tilde{\boldsymbol{P}}\}\right] \leqslant \mathcal{O}(\log\log T) + t_0 + \gamma\left(\mathcal{O}(1) + \mathcal{O}(1) + \mathcal{O}(\log\log T)\right) = \mathcal{O}\left(\log\log T\right)$, which concludes the proof. □

B.5.3 Final Step of the Proof of Theorem 3 (Upper-Bound on the Regret of UniRank Assuming a Total Order on Items)

The proof of Theorem 3 from Lemmas 13 and 14 is mainly based on an appropriate decomposition of the regret.

Proof of Theorem 3. The upper-bound on the expected number of iterations at which UniRank explores while the leader is the optimal partition is given by Lemma 13.

The upper-bound on the expected number of iterations at which the leader is not the optimal partition is given by Lemma 14.

Let now consider the impact of these upper-bounds on the regret of UniRank.

Let remind that $P_c^* = \{c\}$ for $c \in [K]$, $d^* = K + 1$, and $P_{K+1}^* = [L] \setminus [K]$. Therefore, $\mu^* = \mu_{a^*} = \sum_{k=1}^{K} \rho(\boldsymbol{a}^*, k)$, where $\boldsymbol{a}^* \overset{def}{=} (1, 2, \ldots, K)$.

Let first upper-bound the regret suffered at iteration t while the the leader is the

optimal partition:

$$
\begin{aligned}
R_t^* &= \mu^* - \mathbb{E}_{\boldsymbol{a}(t)}\left[\mu_{\boldsymbol{a}(t)} \mid \tilde{\boldsymbol{P}}(t) = \boldsymbol{P}^*\right] \\
&= \sum_{k=1}^{K} \rho(\boldsymbol{a}^*, k) - \mathbb{E}_{\boldsymbol{a}(t)}\left[\rho(\boldsymbol{a}(t), k) \mid \tilde{\boldsymbol{P}}(t) = \boldsymbol{P}^*\right] \\
&= \sum_{k=1}^{K} \mathbb{P}\left(a_k(t) = k \mid \tilde{\boldsymbol{P}}(t) = \boldsymbol{P}^*\right)\left(\rho(\boldsymbol{a}^*, k) - \mathbb{E}_{\boldsymbol{a}(t)}\left[\rho(\boldsymbol{a}(t), k) \mid a_k(t) = k, \tilde{\boldsymbol{P}}(t) = \boldsymbol{P}^*\right]\right) \\
&\quad + \sum_{k=2}^{K} \mathbb{P}\left(a_{k-1}(t) = k \mid \tilde{\boldsymbol{P}}(t) = \boldsymbol{P}^*\right)\left(\rho(\boldsymbol{a}^*, k-1) - \mathbb{E}_{\boldsymbol{a}(t)}\left[\rho(\boldsymbol{a}(t), k-1) \mid a_{k-1}(t) = k, \tilde{\boldsymbol{P}}(t) = \boldsymbol{P}^*\right]\right) \\
&\quad + \sum_{k=2}^{K} \mathbb{P}\left(a_k(t) = k-1 \mid \tilde{\boldsymbol{P}}(t) = \boldsymbol{P}^*\right)\left(\rho(\boldsymbol{a}^*, k) - \mathbb{E}_{\boldsymbol{a}(t)}\left[\rho(\boldsymbol{a}(t), k) \mid a_k(t) = k-1, \tilde{\boldsymbol{P}}(t) = \boldsymbol{P}^*\right]\right) \\
&\quad + \sum_{\ell=K+1}^{L} \mathbb{P}\left(a_K(t) = \ell \mid \tilde{\boldsymbol{P}}(t) = \boldsymbol{P}^*\right)\left(\rho(\boldsymbol{a}^*, k) - \mathbb{E}_{\boldsymbol{a}(t)}\left[\rho(\boldsymbol{a}(t), K) \mid a_K(t) = \ell, \tilde{\boldsymbol{P}}(t) = \boldsymbol{P}^*\right]\right)
\end{aligned}
$$

Let's focus on the first right hand-side term. As the probability of click at position k only depends on the set of items in positions 1 to $k-1$, and as under the condition $a_k(t) = k \wedge \tilde{\boldsymbol{P}}(t) = \boldsymbol{P}^*$, $\boldsymbol{a}(t)$ and $\boldsymbol{a}^*$ have the same set of items in positions 1 to $k-1$, $\rho(\boldsymbol{a}^*, k) = \mathbb{E}_{\boldsymbol{a}(t)}\left[\rho(\boldsymbol{a}(t), k) \mid a_k(t) = k, \tilde{\boldsymbol{P}}(t) = \boldsymbol{P}^*\right]$. Hence that term is equal to 0.

Let now take a look at the second term. By design of UniRank, as $a_{k-1}(t) = k \wedge \tilde{\boldsymbol{P}}(t) = \boldsymbol{P}^*$, there exists c' such that $P_{c'}(t) = \{k-1, k\}$, and

$$
\begin{aligned}
\mathbb{P}\left(a_{k-1}(t) = k \mid \tilde{\boldsymbol{P}}(t) = \boldsymbol{P}^*\right) &= \mathbb{P}\left(a_{k-1}(t) = k, \exists c', P_{c'}(t) = \{k-1, k\} \mid \tilde{\boldsymbol{P}}(t) = \boldsymbol{P}^*\right) \\
&= \frac{1}{2}\mathbb{P}\left(\exists c', P_{c'}(t) = \{k-1, k\} \mid \tilde{\boldsymbol{P}}(t) = \boldsymbol{P}^*\right).
\end{aligned}
$$

Similarly, the third term corresponds to the existence of c' such that $P_{c'}(t) = \{k-1, k\}$, and

$$
\mathbb{P}\left(a_k(t) = k-1 \mid \tilde{\boldsymbol{P}}(t) = \boldsymbol{P}^*\right) = \frac{1}{2}\mathbb{P}\left(\exists c', P_{c'}(t) = \{k-1, k\} \mid \tilde{\boldsymbol{P}}(t) = \boldsymbol{P}^*\right).
$$

By summing both terms, we have to handle

$$
\begin{aligned}
&\frac{1}{2}\mathbb{P}\left(\exists c', P_{c'}(t) = \{k-1, k\} \mid \tilde{\boldsymbol{P}}(t) = \boldsymbol{P}^*\right) \cdot \\
&\left(\rho(\boldsymbol{a}^*, k-1) + \rho(\boldsymbol{a}^*, k) - \mathbb{E}_{\boldsymbol{a}(t)}\left[\rho(\boldsymbol{a}(t), k-1) + \rho(\boldsymbol{a}(t), k) \mid a_{k-1}(t) = k, a_k(t) = k-1, \tilde{\boldsymbol{P}}(t) = \boldsymbol{P}^*\right]\right),
\end{aligned}
$$

which is equal to $\frac{1}{2}\mathbb{P}\left(\exists c', P_{c'}(t) = \{k-1, k\} \mid \tilde{\boldsymbol{P}}(t) = \boldsymbol{P}^*\right)\Delta_k$, where

$$\Delta_k \stackrel{def}{=} \rho(\boldsymbol{a}^*, k-1) + \rho(\boldsymbol{a}^*, k) - \rho((k-1, k) \circ \boldsymbol{a}^*, k-1) - \rho((k-1, k) \circ \boldsymbol{a}^*, k),$$

as the probability of click at any position k' only depends on the set of items in positions 1 to $k'-1$.

Finally, following the same argumentation, the last term is equal to

$$\frac{1}{2}\mathbb{P}\left(\exists c', P_{c'}(t) = \{K, \ell\} \mid \tilde{\boldsymbol{P}}(t) = \boldsymbol{P}^*\right)\Delta_\ell,$$

where $\Delta_\ell \stackrel{def}{=} \rho(\boldsymbol{a}^*, K) - \rho((K, \ell) \circ \boldsymbol{a}^*, K)$.

Overall

$$\begin{aligned} R_t^* &= \sum_{k=2}^{K} \frac{1}{2}\mathbb{P}\left(\exists c', P_{c'}(t) = \{k-1, k\} \mid \tilde{\boldsymbol{P}}(t) = \boldsymbol{P}^*\right)\Delta_k \\ &\quad + \sum_{\ell=K+1}^{L} \frac{1}{2}\mathbb{P}\left(\exists c', P_{c'}(t) = \{K, \ell\} \mid \tilde{\boldsymbol{P}}(t) = \boldsymbol{P}^*\right)\Delta_\ell \\ &= \sum_{k=2}^{L} \frac{1}{2}\mathbb{P}\left(\exists c', P_{c'}(t) = \{\min(k-1, K), k\} \mid \tilde{\boldsymbol{P}}(t) = \boldsymbol{P}^*\right)\Delta_k. \end{aligned}$$

Let finally upper-bound the overall regret.

$$
\begin{aligned}
R(T) &= \sum_{t=1}^{T} \mu^* - \mathbb{E}_{\boldsymbol{a}(t)}\left[\mu_{\boldsymbol{a}(t)}\right] \\
&= \sum_{t=1}^{T} \mathbb{P}\left(\tilde{\boldsymbol{P}}(t) \neq \boldsymbol{P}^*\right)\left(\mu^* - \mathbb{E}_{\boldsymbol{a}(t)}\left[\mu_{\boldsymbol{a}(t)} \mid \tilde{\boldsymbol{P}}(t) \neq \boldsymbol{P}^*\right]\right) \\
&\quad + \sum_{t=1}^{T} \mathbb{P}\left(\tilde{\boldsymbol{P}}(t) = \boldsymbol{P}^*\right)\left(\mu^* - \mathbb{E}_{\boldsymbol{a}(t)}\left[\mu_{\boldsymbol{a}(t)} \mid \tilde{\boldsymbol{P}}(t) = \boldsymbol{P}^*\right]\right) \\
&\leqslant \sum_{t=1}^{T} \mathbb{P}\left(\tilde{\boldsymbol{P}}(t) \neq \boldsymbol{P}^*\right) K \\
&\quad + \sum_{t=1}^{T} \mathbb{P}\left(\tilde{\boldsymbol{P}}(t) = \boldsymbol{P}^*\right) \sum_{k=2}^{L} \frac{1}{2} \mathbb{P}\left(\exists c', P_{c'}(t) = \{\min(k-1, K), k\} \mid \tilde{\boldsymbol{P}}(t) = \boldsymbol{P}^*\right) \Delta_k \\
&\leqslant \mathcal{O}\left(\log\log T\right) \\
&\quad + \sum_{t=1}^{T} \sum_{k=2}^{L} \frac{1}{2} \mathbb{P}\left(\tilde{\boldsymbol{P}}(t) = \boldsymbol{P}^*, \exists c', P_{c'}(t) = \{\min(k-1, K), k, \}\right) \Delta_k \\
&= \mathcal{O}\left(\log\log T\right) \\
&\quad + \sum_{k=2}^{L} \frac{\Delta_k}{2} \sum_{t=1}^{T} \mathbb{P}\left(\tilde{\boldsymbol{P}}(t) = \boldsymbol{P}^*, \exists c', P_{c'}(t) = \{\min(k-1, K), k, \}\right) \\
&\leqslant \mathcal{O}\left(\log\log T\right) \\
&\quad + \sum_{k=2}^{L} \frac{\Delta_k}{2} \left(\frac{16}{\tilde{\delta}_k^* \tilde{\Delta}_k^2} \log T + \mathcal{O}\left(\log\log T\right)\right) \\
&= \sum_{k=2}^{L} \frac{8\Delta_k}{\tilde{\delta}_k^* \tilde{\Delta}_k^2} \log T + \mathcal{O}\left(\log\log T\right) \\
&= \mathcal{O}\left(\frac{L}{\Delta} \log T\right),
\end{aligned}
$$

where for any index $k \geqslant 2$

$$
\tilde{\Delta}_k \overset{def}{=} \tilde{\Delta}_{\min(k-1,K),k} \qquad \textit{and} \qquad \Delta \overset{def}{=} \min_{k \in \{2,\dots,K\}} \frac{\tilde{\delta}_k^* \tilde{\Delta}_k^2}{8\Delta_k},
$$

which concludes the proof.

□

B.6 UniRank's Theoretical Results While Facing State-of-the-Art Click Models

Here, we prove Corollaries 1 and 2 and then discuss the relationship between our upper-bounds and the known lower bounds.

B.6.1 Proof of Corollary 1 (Upper-Bound on the Regret of UniRank when Facing CM* Click Model)

Corollary 3 is a more precise version of Corollary 1. Its proof consists in identifying the gaps $\tilde{\delta}_k^*$, $\tilde{\Delta}_k$, and Δ_k, where k is the index of an item.

Corollary 3 (Facing CM* click model)**.** *Under the hypotheses of Theorem 3, if the user follows CM with probability θ_i to click on item i when it is observed, then for any index $k \geqslant 2$,*

$$\begin{aligned}
\tilde{\delta}_k^* &= (\theta_{k-1} + \theta_k - \theta_{k-1}\theta_k) \prod_{\ell=1}^{k-2} (1 - \theta_\ell) && \text{if } k \leqslant K, \\
\tilde{\delta}_k^* &= \frac{1}{2} (\theta_K + \theta_k) \prod_{\ell=1}^{K-1} (1 - \theta_\ell) && \text{if } k \geqslant K+1, \\
\tilde{\Delta}_k &\geqslant \frac{\theta_{\min(K,k-1)} - \theta_k}{\theta_{\min(K,k-1)} + \theta_k}, \\
\Delta_k &= 0 && \text{if } k \leqslant K, \\
\Delta_k &= (\theta_K - \theta_k) \prod_{\ell=1}^{K-1} (1 - \theta_\ell) && \text{if } k \geqslant K+1.
\end{aligned}$$

Hence, UniRank fulfills

$$\begin{aligned}
R(T) &\leqslant \sum_{k=K+1}^{L} 16 \frac{\theta_K + \theta_k}{\theta_K - \theta_k} \log T + \mathcal{O}(\log \log T) \\
&= \mathcal{O}\left((L-K) \frac{\Theta_K + \theta_{K+1}}{\Theta_K - \theta_{K+1}} \log T \right).
\end{aligned}$$

Proof of Corollary 3. Values $\tilde{\delta}_k^*$ and Δ_k derive from a straightforward computation given CM model.

Let us prove the lower-bound on $\tilde{\Delta}_k$. Let i and j be two items such that $i \neq j$. Let $\boldsymbol{a}$

be a recommendation such that $\mathbb{P}(c_i(t) \neq c_j(t) \mid \boldsymbol{a}(t) = \boldsymbol{a}) > 0$.

Without loss of generality, assume i appears in $\boldsymbol{a}$ in position k, and if j appears in $\boldsymbol{a}$, it is in a position $\ell > k$. Then

$$\tilde{\Delta}_{i,j}(\boldsymbol{a}) = \frac{A\frac{1+B}{2}(\theta_i - \theta_j)}{A\frac{1+B}{2}(\theta_i + \theta_j) - AB\theta_i\theta_j} \geqslant \frac{\theta_i - \theta_j}{\theta_i + \theta_j},$$

with $A \overset{def}{=} \prod_{c=1}^{k-1}(1-\theta_{a_c})$ and $B \overset{def}{=} \prod_{c=k+1}^{\ell-1}(1-\theta_{a_c})$ if j appears in $\boldsymbol{a}$ and 0 otherwise.

Hence the lower-bounding values for $\tilde{\Delta}_k$, by noting that the term A is lower-bounded by $\prod_{\ell=1}^{K-1}(1-\theta_\ell)$.

Regarding the last formula in Lemma 3, it derives from the fact that $\frac{\theta_K+\theta_k}{\theta_K-\theta_k}$ is maximized when θ_k is maximized, meaning $k = K+1$. □

B.6.2 Proof of Corollary 2 (Upper-Bound on the Regret of UniRank when Facing PBM* Click Model)

Corollary 4 is a more precise version of Corollary 2. Its proof consists in identifying the gaps $\tilde{\delta}_k^*$, $\tilde{\Delta}_k$, and Δ_k, where k is the index of an item.

Corollary 4 (Facing PBM* click model)**.** *Under the hypotheses of Theorem 3, if the user follows PBM with the probability θ_i of clicking on item i when it is observed and the probability κ_k of observing the position k, then for any index $k \geqslant 2$,*

$$\begin{aligned}
\tilde{\delta}_k^* &= \frac{1}{2}(\theta_{k-1}+\theta_k)(\kappa_{k-1}+\kappa_k) - 2\theta_{k-1}\theta_k\kappa_{k-1}\kappa_k && \text{if } k \leqslant K,\\
\tilde{\delta}_k^* &= \frac{1}{2}(\theta_K+\theta_k)\kappa_K && \text{if } k \geqslant K+1,\\
\tilde{\Delta}_k &\geqslant \frac{\theta_{\min(K,k-1)} - \theta_k}{\theta_{\min(K,k-1)} + \theta_k},\\
\Delta_k &= (\theta_{k-1}-\theta_k)(\kappa_{k-1}-\kappa_k) && \text{if } k \leqslant K,\\
\Delta_k &= (\theta_K-\theta_k)\kappa_K && \text{if } k \geqslant K+1.
\end{aligned}$$

Hence, UniRank fulfills

$$\begin{aligned}
R(T) &\leqslant \sum_{k=2}^{K} \frac{8(\kappa_{k-1}-\kappa_k)(\theta_{k-1}+\theta_k)^2}{\tilde{\delta}_k^*(\theta_{k-1}-\theta_k)}\log T + \sum_{k=K+1}^{L} 16\frac{\theta_K+\theta_k}{\theta_K-\theta_k}\log T + \mathcal{O}(\log\log T)\\
&= \mathcal{O}\left(\frac{L}{\Delta}\log T\right),
\end{aligned}$$

where $\Delta \stackrel{def}{=} \min\{\min_{k\in\{2,\dots,K\}} \frac{\tilde{\delta}_k^*(\theta_{k-1}-\theta_k)}{(\kappa_{k-1}-\kappa_k)(\theta_{k-1}+\theta_k)^2}, \min_{k\in\{K+1,\dots,L\}} \frac{\theta_K-\theta_k}{\theta_K+\theta_k}\}$.

Proof of Corollary 4. Values $\tilde{\delta}_k^*$ and Δ_k derive from a straightforward computation given PBM model.

Let us prove the lower-bound on $\tilde{\Delta}_k$. Let i and j be two items such that $i \neq j$. Let $\boldsymbol{a}$ be a recommendation such that $\mathbb{P}(c_i(t) \neq c_j(t) \mid \boldsymbol{a}(t) = \boldsymbol{a}) > 0$.

If both i and j appear in $\boldsymbol{a}$, denote $k < \ell$ these positions. Then

$$\tilde{\Delta}_{i,j}(\boldsymbol{a}) = \frac{\frac{1}{2}(\kappa_k + \kappa_\ell)(\theta_i - \theta_j)}{\frac{1}{2}(\kappa_k + \kappa_\ell)(\theta_i + \theta_j) - 2\kappa_k\kappa_\ell\theta_i\theta_j} \geqslant \frac{\theta_i - \theta_j}{\theta_i + \theta_j}.$$

If only one of both items i and j appears in $\boldsymbol{a}$ then $\tilde{\Delta}_{i,j}(\boldsymbol{a}) = \frac{\theta_i-\theta_j}{\theta_i+\theta_j}$.

Hence for any index $k \geqslant 2$, $\tilde{\Delta}_k \geqslant \frac{\theta_{\min(K,k-1)}-\theta_k}{\theta_{\min(K,k-1)}+\theta_k}$. □

UNIVERSITÉ DE RENNES 1

Titre : Recommandation de listes d'items par bandits manchots

Mot clés : Apprentissage en ligne, Systèmes de Recommendations, Bandits Manchots

Résumé : Nous étudions le problème d'apprentissage de l'ordonnancement en ligne de L items pour K positions prédéfinies sur une page web. Pour cela, nous nous intéressons aux algorithmes de bandits manchots qui apprennent les paramètres de modèles de clics identifiés, tel que le modèle basé sur les positions (PBM). Les algorithmes de l'état-de-l'art s'attaquent rarement au PBM complet, où tous les paramètres sont inconnus. De plus, l'état de l'art contient peu d'algorithmes basés sur Thompson Sampling ou sur les bandits unimodaux, malgré leurs performances empiriques reconnues. Nos deux premières contributions s'appuient sur les bandits unimodaux : GRAB est spécialisé pour un PBM complet et UniRank, traite des modèles de clics divers. Ces deux contributions, très efficaces, ont une borne supérieure de regret théorique en $\mathcal{O}\left(L/\Delta \log T\right)$, au niveau de l'état de l'art. La troisième contribution fournit une famille de bandits adressant le problème PBM complet en couplant l'algorithme Thompson Sampling avec des méthodes d'échantillonnage par chaînes de Markov Monte-Carlo (MCMC). Deux méthodes MCMC sont utilisées : par descente de gradient par Langevin, donnant des résultats empiriques semblables à l'état de l'art avec un temps de calcul bas et stable, et par Metropolis Hasting, qui offre le regret empirique le plus bas pour ce problème pour un PBM complet.

Title: List recommendations with multi-armed bandits

Keywords: Online learning, Recommendation Systems, Bandits

Abstract: We tackle the online learning to rank problem of assigning L items to predefined positions on a web page. To address this problem, one can learn, in a multiple-play semi-bandit setting, the parameters of a behavioral click model, e.g. the so-called position-based model (PBM). State-of-the-art algorithms rarely tackle the full PBM, i.e. PBM with all its parameters unknown. Moreover, efficient algorithmic frameworks such as Thompson Sampling or Unimodal bandits were seldom considered for diverse behavioral click models. Three algorithmic contributions are presented in this thesis. Two of them are based on the unimodal bandit setting: GRAB is specialized for full PBM and explores a family of graphs parameterized by the ranking of display positions. UniRank can be used in multiple click models. It builds a graph on partitions of items. These two efficient contributions achieve a theoretical regret upper-bound in $\mathcal{O}\left(L/\Delta \log T\right)$ on par with the state-of-the-art. The third contribution proposes a family of bandit algorithms designed to handle the full PBM and are based on a Thompson Sampling framework, coupled with Markov Chain Monte Carlo (MCMC) sampling methods. Two MCMC methods are used: Langevin Gradient Descent, which shows good empirical regret performance with a low and stable computation time and Metropolis Hasting, less efficient but with the lowest empirical regret seen in the state-of-the-art for so few model assumptions.