MULTI-ARMED BANDITS FOR PREFERENCE LEARNING

by

Sumeet Katariya

A dissertation submitted in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy

(Electrical and Computer Engineering)

at the

UNIVERSITY OF WISCONSIN–MADISON

2018

Date of final oral examination: 12/03/2018

The dissertation is approved by the following members of the Final Oral Committee:
Varun Jog, Assistant Professor, Electrical and Computer Engineering
Robert Nowak, Professor, Electrical and Computer Engineering
Dimitris Papailiopoulos, Assistant Professor, Electrical and Computer Engineering
Garvesh Raskutti, Assistant Professor, Statistics
Stephen Wright, Professor, Computer Science

ACKNOWLEDGMENTS

I feel indebted to many people without whom I would never have reached this milestone. It is impossible to thank all of them, but I would like to acknowledge some special few.

The first of these thanks must go to my advisor, Robert Nowak. I consider myself fortunate to get an opportunity to work with him and observe and learn from him. Rob is a phenomenal researcher. His level of clarity of thought and intuition is something I strive towards to this day. Meetings with him are very productive. He is also a great mentor. With Rob I had the freedom to forge my own research direction, pursue hard problems, and work without feeling rushed. This freedom is necessary in order to be a competent independent researcher.

Brano was never my official mentor, and yet half of my thesis consists of collaborations with him, so I like to think of him as my adopted advisor. Brano helped me grow from a smart engineer to a good researcher who is comfortable with uncertainty. His hands-on style was a perfect complement to Rob's. His enthusiasm, discipline, and optimism make working with him a pleasant experience. Brano is someone I can turn to for any kind of advice, and I am grateful for that. I hope this is just the start of many years of friendship and collaboration.

Graduate school at University of Wisconsin-Madison gave me the opportunity to interact with and learn from some of the best professors. Stark advised me during my Masters and our work resulted in my first publication. I explored interesting problems in optimization with Steve, and causality with Garvesh. My committee members - Dimitris, Varun, Garvesh, Steve, and Rob - gave me valuable feedback on my thesis. I am grateful to all.

The work in this dissertation would not have been possible without my collaborators. I learnt a lot and enjoyed working with Csaba, Mohammad, Kevin, Lalit, Ardhendu, Claire, Zheng, Alan, Yinlam, and Atul.

I thank my labmates Blake, Davis, Urvashi, Scott for the discussions at tea@3, hanabi games, helping me practice talks, and in general making graduate school an enjoyable experience. Friends outside of lab - Dhananjay, Atul, Babbban, Arpit, Mandar, Aastha, Yashdeep, Divya, Indro, Gullu, and many others - made my experience in Madison something I will treasure all my life.

Going back a bit, my journey into the beautiful world of mathematics started thanks to Prof. M. Prakash. I met a great friend and teacher in Udayan, who helped me understand what it means to really understand something.

The last but not the least, thanks are due to my family. I am grateful to Pappa and Mummy for creating a culture at home where pursuit of knowledge was the highest virtue. My brother Namit, although younger, has been an inspiration and guide in more ways than I can remember. Finally, no words are enough to express the gratitude I feel towards Ruchita for all her sacrifices, for always believing in me, and for her boundless love.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

The *multi-armed bandit (MAB)* problem is one of the simplest instances of sequential or adaptive decision making, in which a *learner* needs to select options from a given set of alternatives repeatedly in an online manner. More specifically, the agent selects one option at a time, and observes a numerical (and typically noisy) *reward* signal providing information on the quality of that option, which informs its future selections.

This thesis studies adaptive decision making under different circumstances. The first half of the thesis studies learning using pairwise comparisons. The algorithms depend on the objective of the experimenter. We study the objectives of finding the best item, and approximately ranking the given set of items. In the second half of the thesis, we study the problem of learning from user-clicks. A variety of models have been proposed to simulate user behavior on a search-engine results page, and we study learning in *cold-start* scenarios under two models: the dependent-click model and the position-based model. Finally, if partial prior information about the quality of items is available, we study learning in such *warm-start* circumstances. In these cases, our algorithm provides the experimenter means to control the exploration of the bandit algorithm.

In all cases, we propose algorithms and prove theoretical guarantees about their performance. We also experimentally measure gains with respect to non-adaptive and state-of-the-art adaptive algorithms.

1 INTRODUCTION

The *multi-armed bandit (MAB)* problem is one of the simplest instances of the sequential decision making problem, in which a *learner* needs to select (pull/draw) options from a given set of alternatives (arms) repeatedly in an online manner. More specifically, the agent selects one option at a time, and observes a numerical (and typically noisy) *reward* signal providing information on the quality of that option.

The MAB problem is further classified into two frameworks depending on the goal of the learner. In the *regret minimization* (also called *exploration-exploitation*) framework, the goal of the learner is to minimize its cumulative regret, which is defined as the expected difference between the sum of rewards actually obtained by the learner and the sum of rewards that could have been obtained by playing the best arm in each round. In the *best-arm identification* (also called *pure exploration*) framework, the goal of the learner is to optimally explore the environment so as to identify the best arm. What distinguishes best-arm identification from regret minimization is that in best-arm identification, the exploration phase and evaluation phase are separated.

The difference is best explained by an example that we borrow from Bubeck et al. [2011]. In the case of a severe disease, ill patients only are included in the trial and the cost of picking the wrong treatment is high (the associated reward would equal a large negative value). It is important to minimize the cumulative regret, since the test and cure phases coincide. However, for cosmetic products, there exists a test phase separated from the commercialization phase, and one aims at minimizing the regret of the commercialized product rather than the cumulative regret in the test phase, which is irrelevant.

In this work, the first two problems belong to the best-arm identification framework, while the next four aim to minimize regret.

Organization

Chapters 2 and 3 explore problems in the pairwise-comparison or dueling setting where instead of pulling an arm, the learning agent compares/duels two arms and receives 1-bit feedback about the winner of the duel. This setting is especially suitable when feedback is obtained from humans. The second chapter studies the complexity of finding the best-arm. The third chapter looks at coarsely ranking the arms from best to worst.

The objective in chapters 4-7 is regret minimization. In the fourth chapter, we study online learning models of user interaction with content, for e.g., search results in response to a query. In particular we study the dependent-click model, where the probability of clicking an item decreases as its displayed position increases. In the fifth chapter, we study conservative learning under a simple position-independent click model. One problem with bandit algorithms is that they can explore aggressively especially in the start, and this work provides the experimenter a knob to control the amount of exploration. In chapters 6 and 7, we study the problem of finding the maximum entry in a stochastic rank-1 matrix. This model is inspired from the problem of targeting promotions optimally.

We next briefly summarize the problem studied in each chapter and the results obtained.

1.1 Best-arm Identification

1.1.1 Dueling Bandits with the Borda Voting Rule

Pairwise comparisons (also referred to as the dueling setting) can be an excellent way of collecting information from humans, and from time to time we wish to identify a “most” preferred item by polling the crowd. However, there is no unique way to define the “winner” item. The Condorcet winner, Borda winner, Copeland winner are a few of the many perfectly valid ways to define the winner, and in fact this is a widely studied topic in social choice and voting theory.

In this work, we consider the Borda rule to map pairwise comparisons to a winner, and design an algorithm with this metric in mind. To make this mathematically rigorous, we assume that whenever we query if a is preferred to b, we receive an independent Bernoulli random variable $X_{a,b}$ with expectation $p_{a,b}$. We define the Borda score of item i with respect to the other K objects as $s_i := \frac{1}{K-1}\sum_{j\neq i} p_{i,j}$, so that s_i can be interpreted as the probability that i beats an item chosen uniformly at random from the remaining $[K] \setminus i$ items i.e., $s_i = \mathbb{E}\left[p_{i,J}\right] = \mathbb{E}\left[\mathbb{E}\left[X_{i,j}|J = j\right]\right] J$. The Borda winner is defined as the item with the highest Borda score.

The "Borda dueling bandits" online learning problem can be turned into the standard pure-rewards multi-armed bandit problem where the mean reward of arm i is s_i, and a pull of arm i yields reward $X_{i,J}$ where J is an arm drawn uniformly at random from $[K] \setminus i$. In this work, we explore if this is the best we can do (the answer is yes in the absence of any structural assumptions), and then consider a sparsity structure natural to this problem and show that it results in reduced sample complexity.

1.1.2 Coarse Ranking

This work studies the problem of *coarse ranking*, where the goal is to sort items according to their means into clusters of pre-specified sizes. In many big-data applications, finding the *total* ranking can be infeasible and / or unnecessary, and we may only be interested in the top items, bottom items, or quantiles. Consider for instance the problem of assessing the safety of neighborhoods from pairwise comparisons of Google Streetview images as is done in the PlacePulse project aimed at developing social policy. Finding a complete ordering of the images in this case is impractical because many images are difficult to compare i.e., their safety scores are very close. Furthermore, a total ordering may be unnecessary from a public policy point of view: one may only be interested in the unsafe-appearing images, or the *approximate* rank of every image on the safe-unsafe spectrum.

As another example, consider the problem of assigning grades to students in massive open online courses using peer reviews. Here again, only the *quantile* that every student belongs to is desired (the number of quantiles is determined by the grading scale). A total ordering is unnecessary and may also be infeasible if the number of students is large. The recommender-systems solution of adaptively finding the top items does not help in this case because the grade of *every* student is desired, not just those at the top.

In this work, we propose a computationally efficient PAC algorithm LUCBRank to solve the coarse ranking problem, and derive an upper bound on its sample complexity. We also derive a nearly matching distribution-dependent lower bound.

1.2 Regret Minimization

1.2.1 DCM Bandits: Learning to Rank with Multiple Clicks

Search engines recommend a list of web pages. The user examines this list, from the first page to the last, and may click on multiple attractive pages. The type of user behavior can be modeled by the *dependent-click model (DCM)*. In this work, we propose an online learning variant of this model, which we call *DCM bandits*.

At time t, our learning agent recommends to the user a list of K items. The user examines the items in the list, from the first item to the last. If the examined item attracts the user, the user clicks on it. This is observed by the learning agent. After the user clicks on the item and investigates it, the user decides whether to leave or examine more items. If the user leaves, the DCM interprets this event as that the user is satisfied, and our learning agent receives a reward of one. If the user scans the list of items until the end and does not leave on purpose, the agent receives a reward of zero. The goal of the learning agent is to maximize its total reward, or equivalently to minimize its cumulative regret with respect to the most satisfactory list of K items.

In this work, we propose a computationally efficient algorithm `dcmKL-UCB`, prove gap-dependent upper bounds on the regret of `dcmKL-UCB`, and derive a matching lower bound up to logarithmic factors.

1.2.2 Conservative Exploration using Interleaving

In many practical problems, a learning agent may want to learn the best action in hindsight without ever taking a bad action, which is much worse than a default production action. In general, this is impossible because the agent has to explore unknown actions, some of which can be bad, to learn better actions. However, when the actions are structured, this is possible if the unknown action can be evaluated by interleaving it with the default action. We formalize this concept as learning in stochastic combinatorial semi-bandits with exchangeable actions. We design efficient learning algorithms for this problem, bound their n-step regret, and evaluate them on both synthetic and real-world problems. Our real-world experiments show that our algorithms can learn to recommend K most attractive movies without ever making disastrous recommendations, both overall and subject to a diversity constraint.

1.2.3 Stochastic Rank-1 Bandits

In this work, we study the problem of finding the maximum entry of a stochastic rank-1 matrix from noisy and adaptively-chosen observations. This problem is motivated by ranking in the position-based model (PBM).

The *position-based model* (PBM) is one of the most fundamental click models, a model of how people click on a list of K items out of L. This model is defined as follows. Each *item* is associated with its *attraction* and each *position* in the list is associated with its *examination*. The attraction of any item and the examination of any position are i.i.d. Bernoulli random variables. The item in the list is *clicked* only if it is attractive and its position is examined. Under these assumptions, the pair of the item and position that maximizes the probability of clicking is the maximum entry of a rank-1 matrix, which is the outer product of the attraction probabilities of items and the examination probabilities of positions.

In this work, we propose an online learning model for solving our motivating problem, which we call a *stochastic rank-*1 *bandit*. The learning agent interacts with our problem as follows. At time t, the agent selects a pair of row and column arms, and receives the product of their individual values as a reward. The values are stochastic, drawn independently, and not observed. The goal of the agent is to maximize its expected cumulative reward, or equivalently to minimize its expected cumulative regret with respect to the optimal solution, the most rewarding pair of row and column arms.

We design an elimination algorithm for solving it, which we call `Rank1Elim`. We derive a gap-dependent upper bound on its n-step regret, and a nearly matching gap-dependent lower bound.

1.2.4 Bernoulli Rank-1 Bandits for Click Feedback

The probability that a user will click a search result depends both on its relevance and its position on the results page. The *position based model* explains this behavior by ascribing to every item an *attraction* probability, and to every position an *examination* probability. To be clicked, a result must be both attractive and examined. The probabilities of an item-position pair being clicked thus form the entries of a rank-1 matrix. We propose the learning problem of a *Bernoulli rank-*1 *bandit* where at each step, the learning agent chooses a pair of row and column arms, and receives the product of their Bernoulli-distributed values as a reward. This is a special case of the stochastic rank-1 bandit problem considered in recent work that proposed an elimination based algorithm `Rank1Elim`, and showed that `Rank1Elim`'s regret scales linearly with the number of rows and columns on "benign" instances. These are the instances where the minimum of the average row and column rewards μ is bounded away from zero. The issue with `Rank1Elim` is that it fails to be competitive with straightforward bandit strategies as $\mu \to 0$. In this chapter we propose `Rank1ElimKL`, which replaces the crude confidence intervals of `Rank1Elim` with confidence intervals based on Kullback-Leibler (KL) divergences. With the help of a novel result concerning the scaling of KL divergences we prove that with this change, our algorithm will be competitive no matter the value of μ. Experiments with synthetic data confirm that on benign instances the performance of `Rank1ElimKL` is significantly better than that of even `Rank1Elim`. Similarly, experiments with models derived from real-data confirm that the improvements are significant across the board, regardless of whether the data is benign or not.

2.1 Introduction

The dueling bandit is a variation of the classic multi-armed bandit problem in which the actions are noisy comparisons between arms, rather than observations from the arms themselves [Yue et al., 2012]. Each action provides 1 bit indicating which of two arms is probably better. For example, the arms could represent objects and the bits could be responses from people asked to compare pairs of objects. In this chapter, we focus on the pure *exploration* problem of finding the "best" arm from noisy pairwise comparisons. This problem is different from the *explore-exploit* problem studied in Yue et al. [2012]. There can be different notions of "best" in the dueling framework, including the Condorcet and Borda criteria (defined below).

Most of the dueling-bandit algorithms are primarily concerned with finding the Condorcet winner (the arm that is probably as good or better than every other arm). There are two drawbacks to this. First, a Condorcet winner does not exist unless the underlying probability matrix governing the outcomes of pairwise comparisons satisfies certain restrictions. These restrictions may not be met in many situations. In fact, we show that a Condorcet winner doesn't exist in our experiment with real data presented below. Second, the best known upper bounds on the sample complexity of finding the Condorcet winner (assuming it exists) grow quadratically (at least) with the number of arms. This makes Condorcet algorithms impractical for large numbers of arms.

To address these drawbacks, we consider the Borda criterion instead. The Borda score of an arm is the probability that the arm is preferred to another arm chosen uniformly at random. A Borda winner (arm with the largest Borda score) always exists for every possible probability matrix. We assume throughout this chapter that there exists a unique Borda winner. Finding the Borda winner with probability at least $1-\delta$ can be reduced to solving an instance of the standard multi-armed bandit problem resulting in a sufficient sample complexity of $\mathcal{O}\left(\sum_{i>1}(s_1-s_i)^{-2}\log\left(\log((s_1-s_i)^{-2})/\delta\right)\right)$, where s_i denotes Borda score of arm

i and $s_1 > s_2 > \cdots > s_n$ are the scores in descending order [Karnin et al., 2013, Jamieson et al., 2014]. In favorable cases, for instance, if $s_1 - s_i \geqslant c$, a constant for all $i > 1$, then this sample complexity is linear in n as opposed to the quadratic sample complexity necessary to find the Condorcet winner. In this chapter we show that this upper bound is essentially tight, thereby apparently "closing" the Borda winner identification problem. However, in this chapter we consider a specific type of structure that is motivated by its existence in real datasets that complicates this apparently simple story. In particular, we show that the reduction to a standard multi-armed bandit problem can result in very bad performance when compared to an algorithm that exploits this observed structure.

We explore the sample complexity dependence in more detail and consider structural constraints on the matrix (a particular form of sparsity natural to this problem) that can significantly reduce the sample complexity. The sparsity model captures the commonly observed behavior in elections in which there are a small set of "top" candidates that are competing to be the winner but only differ on a small number of attributes, while a large set of "others" are mostly irrelevant as far as predicting the winner is concerned in the sense that they would always lose in a pairwise matchup against one of the "top" candidates.

This motivates a new algorithm called Successive Elimination with Comparison Sparsity (SECS). SECS takes advantage of this structure by determining which of two arms is better on the basis of their performance with respect to a sparse set of "comparison" arms. Experimental results with real data demonstrate the practicality of the sparsity model and show that SECS can provide significant improvements over standard approaches.

The main contributions of this chapter are as follows:

- A distribution dependent lower bound for the sample complexity of identifying the Borda winner that essentially shows that the Borda reduction to the standard multi-armed bandit problem (explained in detail later) is essentially optimal up to logarithmic factors, given no prior structural information.

- A new structural assumption for the n-armed dueling bandits problem in which the top arms can be distinguished by duels with a sparse set of other arms.

- An algorithm for the dueling bandits problem under this assumption, with theoretical performance guarantees showing significant sample complexity improvements compared to naive reductions to standard multi-armed bandit algorithms.
- Experimental results, based on real-world applications, demonstrating the superior performance of our algorithm compared to existing methods.

2.2 Problem Setup

The *n-armed dueling bandits problem* [Yue et al., 2012] is a modification of the *n-armed bandit problem*, where instead of pulling a single arm, we choose a pair of arms (i, j) to duel, and receive one bit indicating which of the two is better or preferred, with the probability of i winning the duel is equal to a constant $p_{i,j}$ and that of j equal to $p_{j,i} = 1 - p_{i,j}$. We define the *probabilty matrix* $P = [p_{i,j}]$, whose (i, j)th entry is $p_{i,j}$.

Almost all existing n-armed dueling bandit methods [Yue et al., 2012, Yue and Joachims, 2011, Zoghi et al., 2013, Urvoy et al., 2013, Ailon et al., 2014] focus on the explore-exploit problem and furthermore make a variety of assumptions on the preference matrix P. In particular, those works assume the existence of a Condorcet winner: an arm, c, such that $p_{c,j} > \frac{1}{2}$ for all $j \neq c$. The *Borda* winner is an arm b that satisfies $\sum_{j \neq b} p_{b,j} \geqslant \sum_{j \neq i} p_{i,j}$ for all $i = 1, \cdots, n$. In other words, the Borda winner is the arm with the highest average probability of winning against other arms, or said another way, the arm that has the highest probability of winning against an arm selected uniformly at random from the remaining arms. The Condorcet winner has been given more attention than the Borda, the reasons being: 1) Given a choice between the Borda and the Condorcet winner, the latter is preferred in a direct comparison between the two. 2) As pointed out in Urvoy et al. [2013], Zoghi et al. [2013] the Borda winner can be found by reducing the dueling bandit problem to a standard multi-armed bandit problem as follows.

Definition 2.1. Borda Reduction. *The action of pulling arm* i *with reward* $\frac{1}{n-1}\sum_{j\neq i} p_{i,j}$ *can be simulated by dueling arm* i *with another arm chosen uniformly at random.*

However, we feel that the Borda problem has received far less attention than it deserves. Firstly, the Borda winner *always exists*, the Condorcet does not. For example, a Condorcet winner does not exist in the MSLR-WEB10k datasets considered in this chapter. Assuming the existence of a Condorcet winner severely restricts the class of allowed P matrices: only those P matrices are allowed which have a row with all entries $\geqslant \frac{1}{2}$. In fact, Yue et al. [2012], Yue and Joachims [2011] require that the comparison probabilities $p_{i,j}$ satisfy additional transitivity conditions that are often violated in practice. Secondly, there are many cases where the Borda winner and the Condorcet winner are distinct, and the Borda winner would be preferred in many cases. Lets assume that arm c is the Condorcet winner, with $p_{c,i} = 0.51$ for $i \neq c$. Let arm b be the Borda winner with $p_{b,i} = 1$ for $i \neq b, c$, and $p_{b,c} = 0.49$. It is reasonable that arm c is only marginally better than the other arms, while arm b is significantly preferred over all other arms except against arm c where it is marginally rejected. In this example - chosen extreme to highlight the pervasiveness of situations where the Borda arm is preferred - it is clear that arm b should be the winner: think of the arms representing objects being contested such as t-shirt designs, and the P matrix is generated by showing users a pair of items and asking them to choose the better among the two. This example also shows that the Borda winner is more robust to estimation errors in the P matrix (for instance, when the P matrix is estimated by asking a small sample of the entire population to vote among pairwise choices). The Condorcet winner is sensitive to entries in the Condorcet arm's row that are close to $\frac{1}{2}$, which is not the case for the Borda winner. Finally, there are important cases (explained next) where the winner can be found in fewer number of duels than would be required by Borda reduction.

$$P_1 = \begin{array}{c|ccccc|cc} & 1 & 2 & 3 & \cdots & n & s_i & s_1 - s_i \\ 1 & \frac{1}{2} & \frac{1}{2} & \frac{3}{4} & \cdots & \frac{3}{4}+\epsilon & \frac{\frac{1}{2}+\epsilon}{n-1} + \frac{3}{4}\frac{n-2}{n-1} & 0 \\ 2 & \frac{1}{2} & \frac{1}{2} & \frac{3}{4} & \cdots & \frac{3}{4} & \frac{\frac{1}{2}}{n-1} + \frac{3}{4}\frac{n-2}{n-1} & \frac{\epsilon}{n-1} \\ 3 & \frac{1}{4} & \frac{1}{4} & \frac{1}{2} & \cdots & \frac{1}{2} & \frac{1}{2}\frac{n-2}{n-1} & \frac{\frac{1}{2}+\epsilon}{n-1} + \frac{1}{4}\frac{n-2}{n-1} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ n & \frac{1}{4}-\epsilon & \frac{1}{4} & \frac{1}{2} & \cdots & \frac{1}{2} & -\frac{\epsilon}{n-1} + \frac{1}{2}\frac{n-2}{n-1} & \frac{\frac{1}{2}+2\epsilon}{n-1} + \frac{1}{4}\frac{n-2}{n-1} \end{array} \tag{2.1}$$

$$P_2 = \begin{array}{c|ccccc|cc} & 1 & 2 & 3 & \cdots & n & s_i & s_1 - s_i \\ 1 & \frac{1}{2} & \frac{1}{2}+\frac{\epsilon}{n-1} & \frac{3}{4}+\frac{\epsilon}{n-1} & \cdots & \frac{3}{4}+\frac{\epsilon}{n-1} & \frac{\frac{1}{2}+\epsilon}{n-1} + \frac{3}{4}\frac{n-2}{n-1} & 0 \\ 2 & \frac{1}{2}-\frac{\epsilon}{n-1} & \frac{1}{2} & \frac{3}{4} & \cdots & \frac{3}{4} & \frac{\frac{1}{2}-\frac{\epsilon}{n-1}}{n-1} + \frac{3}{4}\frac{n-2}{n-1} & \frac{\epsilon}{n-1} + \frac{\epsilon}{(n-1)^2} \\ 3 & \frac{1}{4}-\frac{\epsilon}{n-1} & \frac{1}{4} & \frac{1}{2} & \cdots & \frac{1}{2} & \frac{-\frac{\epsilon}{n-1}}{n-1} + \frac{1}{2}\frac{n-2}{n-1} & \frac{\frac{1}{2}+\epsilon+\frac{\epsilon}{n-1}}{n-1} + \frac{1}{4}\frac{n-2}{n-1} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ n & \frac{1}{4}-\frac{\epsilon}{n-1} & \frac{1}{4} & \frac{1}{2} & \cdots & \frac{1}{2} & \frac{-\frac{\epsilon}{n-1}}{n-1} + \frac{1}{2}\frac{n-2}{n-1} & \frac{\frac{1}{2}+\epsilon+\frac{\epsilon}{n-1}}{n-1} + \frac{1}{4}\frac{n-2}{n-1} \end{array} \tag{2.2}$$

2.3 Motivation

We define the *Borda score* of an arm i to be the probability of the i^{th} arm winning a duel with another arm chosen uniformly at random:

$$s_i = \tfrac{1}{n-1} \sum_{j \neq i} p_{i,j} \, .$$

Without loss of generality, we assume that $s_1 > s_2 \geqslant \cdots \geqslant s_n$ but that this ordering is unknown to the algorithm. As mentioned above, if the Borda reduction is used then the dueling bandit problem becomes a regular multi-armed bandit problem and lower bounds for the multi-armed bandit problem [Kaufmann et al., 2014, Mannor and Tsitsiklis, 2004] suggest that the number of samples required should scale like $\Omega\left(\sum_{i\neq 1}\frac{1}{(s_1-s_i)^2}\log\frac{1}{\delta}\right)$, which depends only on the Borda scores, and not the individual entries of the preference matrix. This would imply that any preference matrix P with Borda scores s_i is just as hard as another matrix P' with Borda scores s_i' as long as $(s_1 - s_i) = (s_1' - s_i')$. Of course, this lower bound only applies to algorithms using the Borda reduction, and not any algorithm for identifying the Borda winner that may, for instance, collect the duels in a more deliberate way. Next we consider specific P matrices that exhibit two very different kinds of structure but have the same differences in Borda scores which motivates the structure considered in this chapter.

2.3.1 Preference Matrix P known up to permutation of indices

Shown below in equations (2.1) and (2.2) are two preference matrices P_1 and P_2 indexed by the number of arms n that essentially have the same Borda gaps – $(s_1 - s_i)$ is either like $\frac{\epsilon}{n}$ or approximately 1/4 – but we will argue that P_1 is much "easier" than P_2 in a certain sense (assume ϵ is an unknown constant, like $\epsilon = 1/5$). Specifically, if given P_1 and P_2 up to a permutation of the labels of their indices (i.e. given $\Lambda P_1 \Lambda^T$ for some unknown permutation matrix Λ), how many comparisons does it take to find the Borda winner in each case for different values of n?

Recall from above that if we ignore the fact that we know the matrices up to a permutation and use the Borda reduction technique, we can use a multi-armed bandit algorithm (e.g. Karnin et al. [2013], Jamieson et al. [2014]) and find the best arm for both P_1 and P_2 using $O\left(n^2\log(\log(n))\right)$ samples. We next argue that given P_1 and P_2 up to a permutation, there exists an algorithm that can identify the Borda winner of P_1 with just $O(n\log(n))$ samples while the identification of the Borda winner for P_2 requires at least $\Omega(n^2)$ samples. This shows that given the probability matrices up to a permutation, the sample complexity of identifying the Borda winner does not rely just on the Borda differences, but on the particular structure of the probability matrix.

Consider P_1. We claim that there exists a procedure that exploits the structure of the matrix to find the best arm of P_1 using just $O(n \log(n))$ samples. Here's how: For each arm, duel it with $32 \log \frac{n}{\delta}$ other arms chosen uniformly at random. By Hoeffding's inequality, with probability at least $1 - \delta$ our empirical estimate of the Borda score will be within $1/8$ of its true value for all n arms and we can remove the bottom $(n-2)$ arms due to the fact that their Borda gaps exceed $1/4$. Having reduced the possible winners to just two arms, we can identify which rows in the matrix they correspond to and duel each of these two arms against all of the remaining $(n-2)$ arms $O(\frac{1}{\epsilon^2})$ times to find out which one has the larger Borda score using just $O\left(\frac{2(n-2)}{\epsilon^2}\right)$ samples, giving an overall sample complexity of $O\left(n \log n\right)$. We have improved the sample complexity from $O(n^2 \log(\log(n)))$ using the Borda reduction to just $O(n \log(n))$.

Consider P_2. We claim that given this matrix up to a permutation of its indices, no algorithm can determine the winner of P_2 without requesting $\Omega(n^2)$ samples. To see this, suppose an oracle has made the problem easier by reducing the problem down to just the top two rows of the P_2 matrix. This is a binary hypothesis test for which Fano's inequality implies that to guarantee that the probability of error is not above some constant level, the number of samples to identify the Borda winner must scale like $\min_{j \in [n] \setminus \{1,2\}} \frac{1}{KL(p_{1,j}, p_{2,j})} \geqslant \min_{j \in [n] \setminus \{1,2\}} \frac{c}{(p_{1,j} - p_{2,j})^2} = \Omega((n/\epsilon)^2)$ where the inequality holds for some c by the lemma proved in the supplementary materials.

We just argued that the structure of the P matrix, and not just the Borda gaps, can dramatically influence the sample complexity of finding the Borda winner. This leads us to ask the question: if we don't know anything about the P matrix beforehand (i.e. do not know the matrix up to a permutation of its indices), can we learn and exploit this kind of structural information in an online fashion and improve over the Borda reduction scheme? The answer is no, as we argue next.

2.3.2 Distribution-Dependent Lower Bound

We prove a distribution-dependent lower bound on the complexity of finding the best Borda arm for a general P matrix. This is a result important in its own right as it shows that the lower bound obtained for an algorithm using the Borda reduction is tight, that is, this result implies that barring any structural assumptions, the Borda reduction is optimal.

Definition 2.2. δ-PAC dueling bandits algorithm*: A δ-PAC dueling bandits algorithm is an algorithm that selects duels between arms and based on the outcomes finds the Borda winner with probability greater than or equal to* $1 - \delta$*.*

The techniques used to prove the following result are inspired from Lemma 1 in Kaufmann et al. [2014] and Theorem 1 in Mannor and Tsitsiklis [2004].

Theorem 2.3. *(Distribution-Dependent Lower Bound) Consider a matrix* P *such that* $\frac{3}{8} \leqslant p_{i,j} \leqslant \frac{5}{8}, \forall i, j \in [n]$ *with* $n \geqslant 4$*. Let* τ *be the total number of duels. Then for* $\delta \leqslant 0.15$*, any δ-PAC dueling bandits algorithm to find the Borda winner has*

$$\mathbb{E}_P[\tau] \geqslant C \log \frac{1}{2\delta} \sum_{i \neq 1} \frac{1}{(s_1 - s_i)^2}$$

where $s_i = \frac{1}{n-1} \sum_{j \neq i} p_{i,j}$ *denotes the Borda score of arm* i*. Furthermore,* C *can be chosen to be* $1/90$*.*

The proof can be found in the supplementary material.

In particular, this implies that for the preference matrix P_1 in (2.1), any algorithm that makes no assumption about the structure of the P matrix requires $\Omega\left(n^2\right)$ samples. Next we argue that the particular structure found in P_1 is an extreme case of a more general structural phenomenon found in real datasets and that it is a natural structure to assume and design algorithms to exploit.

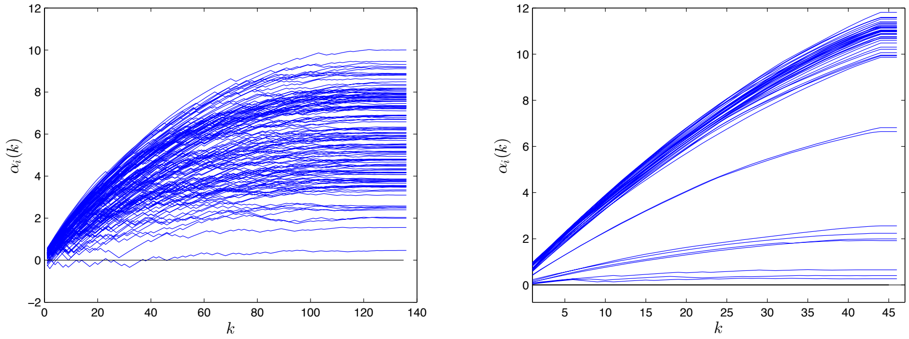
2.3.3 Motivation from Real-World Data

The matrices P_1 and P_2 above illustrate a key structural aspect that can make it easier to find the Borda winner. If the arms with the top Borda scores are distinguished by duels with a small subset of the arms (as exemplified in P_1), then finding the Borda winner may be easier than in the general case. Before formalizing a model for this sort of structure, let us look at two real-world datasets, which motivate the model.

We consider the Microsoft Learning to Rank web search datasets MSLR-WEB10k [Qin et al., 2010] and MQ2008-list [Qin and Liu, 2013] (see the experimental section for a descrptions). Each dataset is used to construct a corresponding probability matrix P. We use these datasets to test the hypothesis that comparisons with a small subset of the arms may suffice to determine which of two arms has a greater Borda score.

Specifically, we will consider the Borda score of the best arm (arm 1) and every other arm. For any other arm $i > 1$ and any positive integer $k \in [n-2]$, let $\Omega_{i,k}$ be a set of cardinality k containing the indices $j \in [n] \setminus \{1, i\}$ with the k largest discrepancies $|p_{1,j} - p_{i,j}|$. These are the duels that, individually, display the greatest differences between arm 1 and i. For each k, define $\alpha_i(k) = 2(p_{1,i} - \frac{1}{2}) + \sum_{j \in \Omega_{i,k}} (p_{1,j} - p_{i,j})$. If the hypothesis holds, then the duels with a small number of (appropriately chosen) arms should indicate that arm 1 is better than arm i. In other words, $\alpha_i(k)$ should become and stay positive as soon as k reaches a relatively small value. Plots of these α_i curves for two datasets are presented in Figures 2.1, and indicate that the Borda winner is apparent for small k. This behavior is explained by the fact that the individual discrepancies $|p_{1,j} - p_{i,j}|$, decay quickly when ordered from largest to smallest, as shown in Figure 2.2.

The take away message is that it is unnecessary to estimate the difference or gap between the Borda scores of two arms. It suffices to compute the *partial* Borda gap based on duels with a small subset of the arms. An appropriately chosen subset of the duels will correctly indicate which arm has a larger Borda score. The algorithm proposed in the next section automatically exploits this structure.



(a) MSLR-WEB10k; n=136

(b) MQ2008; n=46

Figure 2.1: Plots of $\alpha_i(k) = 2(p_{1,i} - \frac{1}{2}) + \sum_{j \in \Omega_{i,k}}(p_{1,j} - p_{1,j})$ vs. k for arms from the (a) MSLR-WEB10k on left (b) MQ2008-list on right. The curves are strictly positive after a small number of duels.

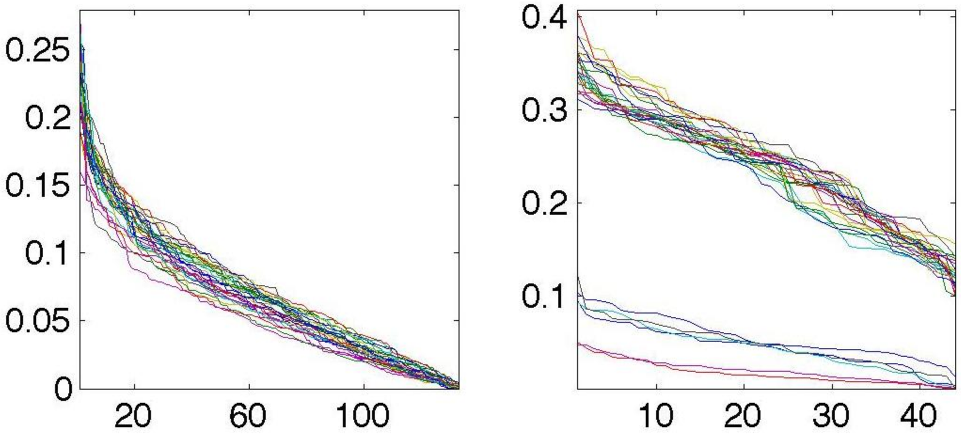


Figure 2.2: Plots of discrepancies $|p_{1,j} - p_{i,j}|$ in descending order for 30 randomly chosen arms (for visualization purposes); MSLR-WEB10k on left, MQ2008-list on right.

2.4 Algorithm and Analysis

In this section we propose a new algorithm that exploits the kind of structure just described above and prove a sample complexity bound. The algorithm is inspired by the Successive Elimination (SE) algorithm of Even-Dar et al. [2006]

Algorithm 1 Sparse Borda Algorithm

```
Input sparsity level $k \in [n-2]$, time gate $T_0 \geqslant 0$.
Start with active set $A_1 = \{1,2,\cdots,n\}$, $t = 1$.
Let $C_t = \sqrt{\frac{2\log(4n^2t^2/\delta)}{t/n}} + \frac{2\log(4n^2t^2/\delta)}{3t/n}$.
while $|A_t| > 1$ do
    Choose $I_t$ uniformly at random $[n]$.
    for $j \in A_t$ do
        Observe $Z_{j,I_t}^{(t)}$ and update $\widehat{p}_{j,I_t,t} = \frac{n}{t}\sum_{\ell=1}^{t} Z_{j,I_\ell}^{(\ell)}\mathbf{1}_{I_\ell=I_t}$, $\widehat{s}_{j,t} = \frac{n/(n-1)}{t}\sum_{\ell=1}^{t} Z_{j,I_\ell}^{(\ell)}$.
    end for
    $A_{t+1} = A_t \setminus \Big\{ j \in A_t : \exists i \in A_t$ with
                1) $\mathbf{1}_{\{t>T_0\}}\widehat{\Delta}_{i,j,t}\Big(\arg\max_{\Omega\subset[n]:|\Omega|=k}\widehat{\nabla}_{i,j,t}(\Omega)\Big) > 6(k+1)C_t$
            OR  2) $\widehat{s}_{i,t} > \widehat{s}_{j,t} + \frac{n}{n-1}\sqrt{\frac{2\log(4nt^2/\delta)}{t}}\Big\}$
    $t \leftarrow t+1$
end while
```

for standard multi-armed bandit problems. Essentially, the proposed algorithm below implements SE with the Borda reduction and an additional elimination criterion that exploits sparsity (condition 1 in the algorithm). We call the algorithm Successive Elimination with Comparison Sparsity (SECS).

We will use $\mathbf{1}_E$ to denote the indicator of the event E and $[n] = \{1,2,\ldots,n\}$. The algorithm maintains an active set of arms A_t such that if $j \notin A_t$ then the algorithm has concluded that arm j is not the Borda winner. At each time t, the algorithm chooses an arm I_t uniformly at random from $[n]$ and compares it with all the arms in A_t. Note that $A_k \subseteq A_\ell$ for all $k \geqslant \ell$. Let $Z_{i,j}^{(t)} \in \{0,1\}$ be independent Bernoulli random variables with $\mathbb{E}[Z_{i,j}^{(t)}] = p_{i,j}$, each denoting the outcome of "dueling" $i, j \in [n]$ at time t (define $Z_{i,j}^{(t)} = 0$ for $i = j$). For any $t \geqslant 1$, $i \in [n]$, and $j \in A_t$ define

$$\widehat{p}_{j,i,t} = \frac{n}{t}\sum_{\ell=1}^{t} Z_{j,I_\ell}^{(\ell)}\mathbf{1}_{I_\ell=i}$$

so that $\mathbb{E}\left[\widehat{p}_{j,i,t}\right] = p_{j,i}$. Furthermore, for any $t \geqslant 1$, $j \in A_t$ define

$$\widehat{s}_{j,t} = \frac{n/(n-1)}{t} \sum_{\ell=1}^{t} Z_{j,I_\ell}^{(\ell)}$$

so that $\mathbb{E}\left[\widehat{s}_{j,t}\right] = s_j$. For any $\Omega \subset [n]$ and $i, j \in [n]$ define

$$\begin{aligned}
\Delta_{i,j}(\Omega) &= 2(p_{i,j} - \tfrac{1}{2}) + \sum_{\omega \in \Omega : \omega \neq i \neq j} (p_{i,\omega} - p_{j,\omega}) \\
\widehat{\Delta}_{i,j,t}(\Omega) &= 2(\widehat{p}_{i,j,t} - \tfrac{1}{2}) + \sum_{\omega \in \Omega : \omega \neq i \neq j} (\widehat{p}_{i,\omega,t} - \widehat{p}_{j,\omega,t}) \\
\nabla_{i,j}(\Omega) &= \sum_{\omega \in \Omega : \omega \neq i \neq j} |p_{i,\omega} - p_{j,\omega}| \\
\widehat{\nabla}_{i,j}(\Omega) &= \sum_{\omega \in \Omega : \omega \neq i \neq j} |\widehat{p}_{i,\omega,t} - \widehat{p}_{j,\omega,t}| \,.
\end{aligned}$$

The quantity $\Delta_{i,j}(\Omega)$ is the *partial* gap between the Borda scores for i and j, based on only the comparisons with the arms in Ω. Note that $\frac{1}{n-1}\Delta_{i,j}([n]) = s_i - s_j$. The quantity $\arg\max_{\Omega \subset [n] : |\Omega| = k} \nabla_{i,j}(\Omega)$ selects the indices ω yielding the largest discrepancies $|p_{i,\omega} - p_{j,\omega}|$. $\widehat{\Delta}$ and $\widehat{\nabla}$ are empirical analogs of these quantities.

Definition 2.4. *For any $i \in [n] \setminus 1$ we say the set $\{(p_{1,\omega} - p_{i,\omega})\}_{\omega \neq 1 \neq i}$ is (γ, k)-approximately sparse if*

$$\max_{\Omega \in [n] : |\Omega| \leqslant k} \nabla_{1,i}(\Omega \setminus \Omega_i) \;\leqslant\; \gamma \Delta_{1,i}(\Omega_i)$$

where $\Omega_i = \arg \max_{\Omega \subset [n] : |\Omega| = k} \nabla_{1,i}(\Omega)$.

Instead of the strong assumption that the set $\{(p_{1,\omega} - p_{i,\omega})\}_{\omega \neq 1 \neq i}$ has no more than k non-zero coefficients, the above definition relaxes this idea and just assumes that the absolute value of the coefficients outside the largest k are small relative to the partial Borda gap. This definition is inspired by the structure described in previous sections and will allow us to find the Borda winner faster.

The parameter T_0 is specified (see Theorem 2.5) to guarantee that all arms with sufficiently large gaps $s_1 - s_i$ are eliminated by time step T_0 (condition 2). Once $t > T_0$, condition 1 also becomes active and the algorithm starts removing arms with large partial Borda gaps, exploiting the assumption that the top arms can be distinguished by comparisons with a sparse set of other arms. The algorithm terminates when only one arm remains.

Theorem 2.5. *Let $k \geqslant 0$ and $T_0 > 0$ be inputs to the above algorithm and let R be the solution to $\frac{32}{R^2} \log\left(\frac{32n/\delta}{R^2}\right) = T_0$. If for all $i \in [n] \setminus 1$, at least one of the following holds:*

1. *$\{(p_{1,\omega} - p_{i,\omega})\}_{\omega \neq 1 \neq i}$ is $(\frac{1}{3}, k)$-approximately sparse,*
2. *$(s_1 - s_i) \geqslant R$,*

then with probability at least $1 - 3\delta$, the algorithm returns the best arm after no more than

$$c \sum_{j>1} \min\left\{ \max\left\{ \tfrac{1}{R^2} \log\left(\tfrac{n/\delta}{R^2}\right), \tfrac{(k+1)^2/n}{\Delta_j^2} \log\left(\tfrac{n/\delta}{\Delta_j^2}\right) \right\}, \tfrac{1}{\Delta_j^2} \log\left(\tfrac{n/\delta}{\Delta_j^2}\right) \right\}$$

samples where $\Delta_j := s_1 - s_j$ and $c > 0$ is an absolute constant.

The second argument of the min is precisely the result one would obtain by running Successive Elimination with the Borda reduction [Even-Dar et al., 2006]. Thus, under the stated assumptions, the algorithm never does worse than the Borda reduction scheme. The first argument of the min indicates the potential improvement gained by exploiting the sparsity assumption. The first argument of the max is the result of throwing out the arms with large Borda differences and the second argument is the result of throwing out arms where a *partial* Borda difference was observed to be large.

To illustrate the potential improvements, consider the P_1 matrix discussed above, the theorem implies that by setting $T_0 = \frac{32}{R^2} \log\left(\frac{32n/\delta}{R^2}\right)$ with $R = \frac{1/2+\epsilon}{n-1} + \frac{1}{4}\frac{n-2}{n-1} \approx \frac{1}{4}$ and $k = 1$ we obtain a sample complexity of $O(\epsilon^{-2} n \log(n))$ for the proposed algorithm compared to the standard Borda reduction sample complexity of $\Omega(n^2)$.

In practice it is difficult optimize the choice of T_0 and k, but motivated by the results shown in the experiments section, we recommend setting $T_0 = 0$ and $k = 5$ for typical problems.

2.5 Experiments

The goal of this section is not to obtain the best possible sample complexity results for the specified datasets, but to show the *relative* performance gain of exploiting structure using the proposed SECS algorithm with respect to the Borda reduction. That is, we just want to measure the effect of exploiting sparsity while keeping all other parts of the algorithms constant. Thus, the algorithm we compare to that uses the simple Borda reduction is simply the SECS algorithm described above but with $T_0 = \infty$ so that the sparse condition never becomes activated. Running the algorithm in this way, it is very closely related to the Successive Elimination algorithm of Even-Dar et al. [2006]. In what follows, our proposed algorithm will be called SECS and the benchmark algorithm will be denoted as just the Borda reduction (BR) algorithm.

We experiment on both simulated data and two real-world datasets. During all experiments, both the BR and SECS algorithms were run with $\delta = 0.1$. For the SECS algorithm we set $T_0 = 0$ to enable condition 1 from the very beginning (recall for BR we set $T_0 = \infty$). Also, while the algorithm has a constant factor of 6 multiplying $(k+1)C_t$, we feel that the analysis that led to this constant is very loose so in practice we recommend the use of a constant of $1/2$ which was used in our experiments. While the change of this constant invalidates the guarantee of Theorem 2.5, we note that in all of the experiments to be presented here, neither algorithm ever failed to return the best arm. This observation also suggests that the SECS algorithm is robust to possible inconsistencies of the model assumptions.

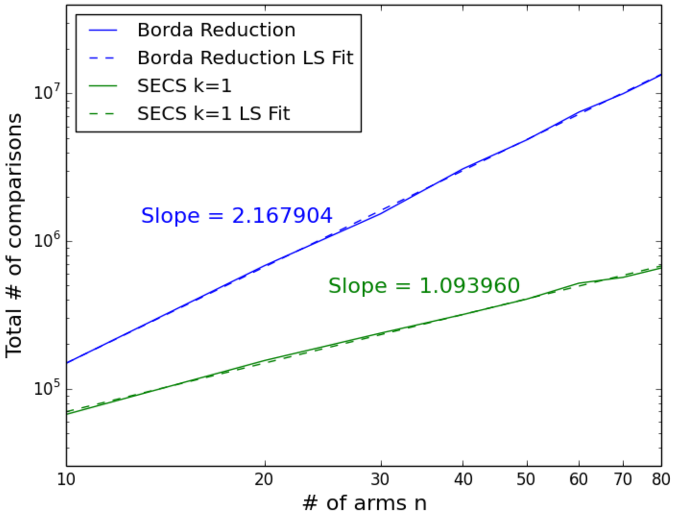


Figure 2.3: Comparison of the Borda reduction algorithm and the proposed SECS algorithm ran on the P_1 matrix for different values of n. Plot is on log-log scale so that the sample complexity grows like n^s where s is the slope of the line.

2.5.1 Synthetic Preference matrix

Both algorithms were tasked with finding the best arm using the P_1 matrix of (2.1) with $\epsilon = 1/5$ for problem sizes equal to $n = 10, 20, 30, 40, 50, 60, 70, 80$ arms. Inspecting the P_1 matrix, we see that a value of $k = 1$ in the SECS algorithm suffices so this is used for all problem sizes. The entries of the preference matrix $P_{i,j}$ are used to simulate comparisons between the respective arms and each experiment was repeated 75 times.

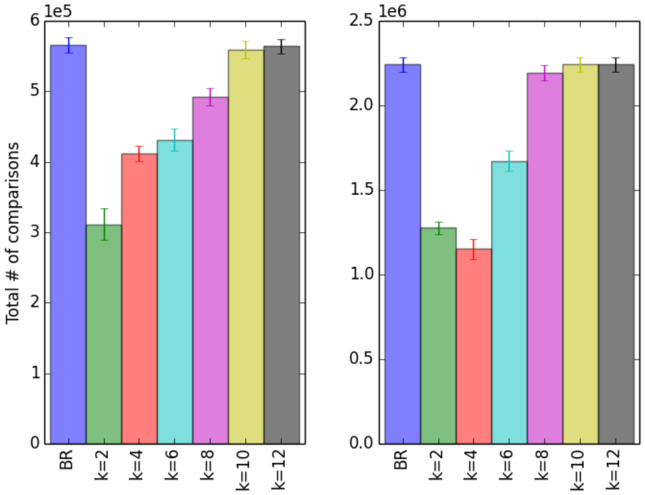
Recall from Section 2.3 that any algorithm using the Borda reduction on the P_1 matrix has a sample complexity of $\Omega(n^2)$. Moreover, inspecting the proof of Theorem 2.5 one concludes that the BR algorithm has a sample complexity of $O(n^2 \log(n))$ for the P_1 matrix. On the other hand, Theorem 2.5 states that the SECS algorithm should have a sample complexity no worse than $O(n \log(n))$ for the P_1 matrix. Figure 2.3 plots the sample complexities of SECS and BR on a log-log plot. On this scale, to match our sample complexity hypotheses, the slope of the BR line should be about 2 while the slope of the SECS line should be about 1, which is exactly what we observe.

2.5.2 Web search data

We consider two web search data sets. The first is the MSLR-WEB10k Microsoft Learning to Rank data set [Qin et al., 2010] that is characterized by approximately 30,000 search queries over a number of documents from search results. The data also contains the values of 136 features and corresponding user labelled relevance factors with respect to each query-document pair. We use the training set of Fold 1, which comprises of about 2,000 queries. The second data set is the MQ2008-list from the Microsoft Learning to Rank 4.0 (MQ2008) data set [Qin and Liu, 2013]. We use the training set of Fold 1, which has about 550 queries. Each query has a list of documents with 46 features and corresponding user labelled relevance factors.

For each data set, we create a set of rankers, each corresponding to a feature from the feature list. The aim of this task is be to determine the feature whose ranking of query-document pairs is the most relevant. To compare two rankers, we randomly choose a pair of documents and compare their relevance rankings with those of the features. Whenever a mismatch occurs between the rankings returned by the two features, the feature whose ranking matches that of the relevance factors of the two documents "wins the duel". If both features rank the documents similarly, the duel is deemed to have resulted in a tie and we flip a fair coin. We run a Monte Carlo simulation on both data sets to obtain a preference matrix P corresponding to their respective feature sets. As with the previous setup, the entries of the preference matrices ($[P]_{i,j} = p_{i,j}$) are used to simulate comparisons between the respective arms and each experiment was repeated 75 times.

From the MSLR-WEB10k data set, a single arm was removed for our experiments as its Borda score was unreasonably close to the arm with the best Borda score and behaved unlike any other arm in the dataset with respect to its α_i curves, confounding our model. For these real datasets, we consider a range of different k values for the SECS algorithm. As noted above, while there is no guarantee that the



(a) MSLR-WEB10k

(b) MQ2008

Figure 2.4: Comparison of an action elimination-style algorithm using the Borda reduction (denoted as BR) and the proposed SECS algorithm with different values of k on the two datasets.

SECS algorithm will return the true Borda winner, in all of our trials for all values of k reported we never observed a single error. This is remarkable as it shows that the correctness of the algorithm is insensitive to the value of k on at least these two real datasets. The sample complexities of BR and SECS on both datasets are reported in Figure 2.4. We observe that the SECS algorithm, for small values of k, can identify the Borda winner using as few as *half* the number required using the Borda reduction method. As k grows, the performance of the SECS algorithm becomes that of the BR algorithm, as predicted by Theorem 2.5.

Lastly, the preference matrices of the two data sets support the argument for finding the Borda winner over the Condorcet winner. The MSLR-WEB10k data set has no Condorcet winner arm. However, while the MQ2008 data set has a Condorcet winner, when we consider the Borda scores of the arms, it ranks second.

3 COARSE RANKING

3.1 Introduction

We consider the problem of efficiently sorting items according to their means into clusters of pre-specified sizes, which we refer to as *coarse ranking*. In many big-data applications, finding the *total* ranking can be infeasible and/or unnecessary, and we may only be interested in the top items, bottom items, or quantiles. Consider for instance the problem of assessing the safety of neighborhoods from pairwise comparisons of Google street view images, as is done in the Place Pulse project [Naik et al., 2014], which can be applied to develop social policy [Dubey et al., 2016]. Finding a complete ordering of the images in this case is impractical because many images are difficult to compare i.e., their safety scores are very close (see Section 3.7.2). Furthermore, a total ordering may be unnecessary from a public policy point of view, since the *approximate* rank of every image on the safe-unsafe spectrum may suffice.

Motivated by these applications, we model the coarse ranking problem as follows. Given K random variables, $c \geqslant 2$ clusters, and cluster boundaries $1 \leqslant \kappa_1 < \kappa_2 < \cdots < \kappa_{c-1} < \kappa_c = K$, the goal is to reliably identify the κ_1 random variables with the highest means, the $\kappa_2 - \kappa_1$ random variables with the highest means among the remaining $K - \kappa_1$ random variables, and so on, by observing samples from their reward distributions (for a precise formulation see Section 3.4). The focus of this chapter is on algorithms that achieve this clustering by requesting samples adaptively. The coarse ranking setting applies to the scenarios above, and also subsumes many well-studied problems. The problem of finding the best item corresponds to $\kappa_1 = 1, \kappa_2 = K$. The problem of finding the top-m items corresponds to $\kappa_1 = m, \kappa_2 = K$. The problem of sorting the items into c equal-sized clusters corresponds to $\kappa_i = \text{round}(iK/c), 1 \leqslant i \leqslant c$. Finally, the complete ranking can be obtained by setting $\kappa_i = i, 1 \leqslant i \leqslant i \leqslant K$.

The problem of *completely sorting* items is in general hard in real-world applications, and does not exhibit gains from adaptivity. Maystre and Grossglauser [2017] who analyze the performance of Quicksort, observe in their real-world experiments:

> "The improvement is noticeable but modest. We notice that item parameters are close to each other on average; . . . This is because there is a considerable fraction of items that have their parameters (means) very close to one another . . . Figuring out the exact order of these images is therefore difficult and probably of marginal value." (3.1)

The fact that *adaptivity doesn't help for complete ranking* is true not just for Quicksort, but other adaptive algorithms as well - as we observe in our experiments. Adaptivity does however help for coarse ranking, and this can be explained. Consider the case when the K items have bounded reward distributions, and their means are equally separated, with a gap Δ between consecutive means. Correctly ordering any two consecutive items requires $\Omega(1/\Delta^2)$ samples, and thus *any* algorithm would require $\Omega(K/\Delta^2)$ to find a total ordering. A non-adaptive algorithm sampling the items uniformly would gather approximately equal samples from every item, and hence will find the correct ranking after roughly these many samples (up to perhaps log factors). Thus adaptivity doesn't help in this case. However, if the goal is to find only the quartiles say, an adaptive algorithm can quickly stop sampling items that are far from the quartile boundaries and gain over non-adaptive algorithms.

In this work, we make six contributions. First, we motivate the coarse ranking setting. We do this by arguing that most real-life problems have high noise, and by explaining why adaptive methods are ineffective in producing a complete ranking in these high-noise regimes (Section 3.3). Second, we precisely formulate the online probably approximately correct (PAC)-coarse ranking problem with error tolerance ϵ and failure probability δ that can model real-valued as well as pairwise comparison feedback (Section 3.4). Third, we propose a nonparametric PAC Upper Confidence Bound (UCB)-type algorithm `LUCBRank` to solve this problem. To the best of our knowledge, this is the first UCB-type algorithm for ranking (Section 3.5). Fourth, we analyze the sample complexity of `LUCBRank` and prove an upper bound which is inversely proportional to the distance of the item to its closest cluster boundary, where the distance is measured in terms of Chernoff information

(Section 3.6). Fifth, we also prove a nearly matching distribution-dependent lower bound. The contribution of an item to the lower bound is inversely proportional to the distance of the item to the closest item in an adjacent cluster, with distance in this case measured using KL-divergences (Section 3.6.3). Finally, we compare the performance of our algorithm to several baselines on synthetic as well as real-world data gathered using MTurk, and observe that it performs 2 - 3x better than existing algorithms even when they have the advantage of knowing the underlying parametric model (Section 3.7).

3.1.1 Ranking using Pairwise Comparisons

We use the term direct-feedback or real-rewards to indicate a setting where the learner can sample directly from the item's reward distribution. Our algorithm is stated for this setting. In contrast, in the pairwise-comparison or dueling setting, the learner compares two items and receives 1-bit feedback about who won the duel. We next explain how to translate our algorithm to this setting.

Any algorithm designed to solve the direct-feedback coarse ranking problem can also be used with pairwise comparison feedback using Borda reduction [Jamieson et al., 2015b]. According to this technique, whenever the algorithm asks to draw a sample from item i, we compare item i to a randomly chosen item j, and ascribe a reward of 1 to item i if i wins the duel, and 0 otherwise. This is equivalent to the rewards being sampled from a Bernoulli distribution with means given by the Borda scores of the items. The Borda score of an item i is defined as

$$p_i := \frac{1}{K-1} \sum_{j \neq i} \mathbb{P}(i > j). \tag{3.2}$$

3.2 Related Work

There is extensive work on ranking from noisy pairwise comparisons, we refer the reader to excellent surveys by Busa-Fekete and Hüllermeier [2014], Agarwal [2016]. We discuss the most relevant work next.

3.2.1 Ranking from Pairwise Comparisons

The pairwise comparison matrix P (where $P_{ij} = \mathbb{P}(i > j)$) and assumptions on it play a major role in the design of ranking algorithms [Agarwal, 2016]. A sequence of progressively relaxed assumptions on P can be shown where ranking methods that work under restrictive assumptions fail when these assumptions are relaxed [Rajkumar and Agarwal, 2014, Rajkumar et al., 2015]. Spectral ranking algorithms have been proposed when comparisons are available for a fixed set of pairs [Negahban et al., 2012a,b]; this corresponds to a partially observed P matrix. Braverman and Mossel [2009], Wauthier et al. [2013] propose and analyze algorithms for the noisy-permutation model; this corresponds to a P matrix which has two types of entries: $1 - p$ in the upper triangle and p in the lower triangle (assuming the true ordering of the items is $1 \dots K$). They also focus on settings where queries cannot be repeated. Our work makes no assumptions on the P matrix and ranks items using their Borda scores. This is important given the futility of parametric models to model real-life scenarios [Shah et al., 2016].

Quicksort is another highly recommended algorithm for ranking using noisy pairwise comparisons. Maystre and Grossglauser [2017] study Quicksort under the BTL noise model, and Alonso et al. [2003] analyze Quicksort under the noisy permutation model. We comment on these in Section 3.3.

Jamieson and Nowak [2011] propose an algorithm for active ranking from pairwise comparisons when points can be embedded in Euclidean space. Ailon [2012] consider ranking when query responses are fixed. More recently, Agarwal et al. [2017] consider top-m item identification and ranking under limited rounds of adaptivity, Falahatgar et al. [2017] consider the problem of finding the maximum and ranking assuming strong-stochastic transitivity and the stochastic-triangle inequality. We do not need these assumptions.

Our setting is closest to the setting proposed by Heckel et al. [2016], in the context of ranking using pairwise comparisons. Our setting however applies to real-valued rewards as well as pairwise comparison feedback. Furthermore, our setting incorporates the notion of ϵ-optimality which allows the user to specify an error tolerance [Even-Dar et al., 2006]. This is important in practice if the item means are very close to each other. Finally, as they note, their Active Ranking (AR) algorithm is an elimination-style algorithm, our `LUCBRank` is UCB-style; it is known that the latter perform better in practice [Jiang et al., 2017]. We also verify this empirically in Section 3.7.2, and observe that `LUCBRank` requires 2-3x fewer samples than AR in our synthetic as well as real-world experiments (see Fig. 3.2 and Fig. 3.5).

3.2.2 Relation to Bandits

The idea of sampling items based on lower and upper confidence bounds is well-known in the bandits literature [Auer, 2002]. However, these algorithms either focus on finding the best or top-m items [Audibert and Bubeck, 2010, Kalyanakrishnan et al., 2012, Kaufmann et al., 2015, Chen et al., 2017], or on minimizing regret [Bubeck et al., 2012]. This is the first work to our knowledge that employs this tool for ranking.

3.3 Motivation

We argue that existing adaptive methods offer no significant gains over their non-adaptive counterparts when the goal is to find a complete ranking, and coarse ranking is more appropriate for many real-world applications. We provide brief theoretical justification for this claim in the discussion after quote (3.1), and empirically verify this behavior in Fig. 3.4. In this section, we focus on Quicksort, because it has been well-studied under multiple noise models. Quicksort has optimal sample complexity when comparisons are noiseless [Sedgewick and Wayne, 2011] and is naturally appealing when comparisons are noisy [Maystre and Grossglauser, 2017]. Intuitively it feels like the right thing to do - by comparing an item with the pivot and putting it left or right appropriately, Quicksort performs a binary search for the true position of an item. However it is far from optimal under two noise models as we argue next.

First, consider the noisy-permutation (NP) noise model [Feige et al., 1994] where the outcomes of pairwise comparisons are independently flipped with an error probability p. In the first stage of Quicksort, every item that is compared with the pivot and put in the wrong bucket contributes on average $\frac{K}{2}$ to the Kendall tau error (total number of inverted pairs). Now, Kp items are put in the wrong bucket on average in the first stage of Quicksort, and hence the total number of inverted pairs is at least $\Omega(K^2p)$. Alonso et al. [2003] show that $\Theta(K^2p)$ is indeed the expected number of inversions. This is far from optimal because Braverman and Mossel [2009] propose an algorithm which has a Kendall tau error of $O(K)$ *with high probability*, using $K \log K$ comparisons (same as quicksort). Alonso et al. [2003] conjecture that for quicksort to have $O(K)$ expected inversions, p needs to go down faster than $1/K$, like $\frac{1}{K \log K}$. As the above calculation shows, they conjecture that this is because Quicksort is extremely brittle: "the main contribution (to the total inversions) comes from the 'first' error, in some sense." One may be able to get rid of this lack of robustness by repeating queries, but this requires knowledge of the error probability p or adapting to its unknown value. This is possible, but as we argue shortly, a good model for real-world problems where comparisons are made by humans is one where p increases to $1/2$ as K grows, since it becomes more difficult to compare adjacent items in the true ranking as K increases. Quicksort certainly fails in this regime.

The other class of well-studied noise models are the Bradley-Terry-Luce (BTL) [Bradley and Terry, 1952] or Thurstone [Thurstone, 1927] models, which assume a K-dimensional weight vector that measures the quality of each item, and the pairwise comparison probabilities are determined via some fixed function of the qualities of pair of objects. These models are more realistic than the NP model since under these models, comparisons between items that are far apart in the true ranking are less noisy than those between nearby items. Maystre and Grossglauser [2017] analyze the expected number of inversions of Quicksort under the BTL model, and show that when the average gap between adjacent items is Δ, the expected number of inversions is $O(\Delta^{-3})$. They note however that real-world datasets have extremely small Δ ($\hat{\Delta}^{-1} = 376$ in their experiments) and Quicksort performs no better than random (see quote (3.1)). We make similar observations about the inefficacy of Quicksort (and other adaptive algorithms) in our real-world experiments (see Fig. 3.4).

The problem in finding an exact/total ranking is that if the means of the items lie in a bounded range, e.g., $[0, 1]$, then the minimum gap must decrease at least linearly with K and many items become essentially indistiguishable. To see this, suppose there is a constant gap Δ between consecutive means and let $m = \lceil \frac{3}{\Delta} \rceil$. Then, assuming the logistic model, the m-th item beats the 1st item with probability $\geqslant 0.95$, the $2m$-th item beats the m-th item with probability $\geqslant 0.95$, and so on. Thus, items that are m-apart can be considered distinguishable. Assuming the range of possible means is bounded implies that $\Delta \propto 1/K$. Thus, the number of items that are essentially indistiguishable increases linearly with K, suggesting that seeking a total ranking is a futile effort. This situation arises in applications such as Place Pulse where humans rate street view images according to their perceived safety [Naik et al., 2014], or the task in Wood et al. [2017] where humans rate face images according to the strength of their emotions.

Coarse ranking allows the experimenter to set the number of clusters in accordance with the number of distinguishable levels, and thus frees the algorithm from the task of distinguishing incomparable items. In this sense, it converts a high-noise problem to a low-noise one. Even though the gap between adjacent items is small, most items are far from their nearest cluster boundary, and an adaptive algorithm can stop sampling these items early.

3.4 Setting

In this section, we precisely formulate the coarse ranking setting. For ease of reference, we use terminology from the bandits literature and refer to an item as arm. Also, pulling or drawing an arm is equivalent to sampling from the item's reward distribution.

Consider a multi-armed bandit with K arms. Each arm a corresponds to a Bernoulli distribution with an unknown mean p_a, denoted $\mathcal{B}(p_a)$. A draw / pull of arm a yields a reward from distribution $\mathcal{B}(p_a)$. Without loss of generality, assume the arms are numbered so that $p_1 \geqslant p_2 \cdots \geqslant p_K$.

Given an integer $c \geqslant 2$ representing the number of clusters, let $1 \leqslant \kappa_1 < \kappa_2 < \cdots < \kappa_c = K$ be a collection of positive integers. Any such collection of positive integers defines a partition of $[K]$ into c disjoint sets of the form

$$M_1^* \coloneqq \{1, \ldots, \kappa_1\},\ M_2^* \coloneqq \{\kappa_1 + 1, \ldots, \kappa_2\}, \ldots, M_c^* \coloneqq \{\kappa_{c-1} + 1, \ldots, K\}. \tag{3.3}$$

To solve the *coarse ranking* problem given a set of cluster boundaries $(\kappa_i)_{i=1}^c$, an algorithm may sample arms of the K-armed bandit and record the results; the algorithm is required to terminate and cluster the arms into an ordered set of disjoint sets of the form (3.3). We refer to this output as a *coarse ranking*.

We next define the notion of ϵ-tolerance. For some fixed tolerance $\epsilon \in [0, 1]$ and $1 \leqslant i \leqslant c$, let $M_{i,\epsilon}^*$ be the set of all arms that should be in cluster i upto a tolerance ϵ, i.e.

$$M_{i,\epsilon}^* \coloneqq \{a : p_{\kappa_{i-1}+1} + \epsilon \geqslant p_a \geqslant p_{\kappa_i} - \epsilon\},$$

(with the convention that $p_{\kappa_0} = 1$). Note that the true set of arms in cluster i: $M_i^* \coloneqq \{\kappa_{i-1} + 1, \ldots, \kappa_i\}$, is a subset of $M_{i,\epsilon}^*$; the latter set contains in addition arms that are ϵ close to the boundary.

For a given mistake probability $\delta \in [0, 1]$ and a given error tolerance $\epsilon \in [0, 1]$, we call an algorithm (ϵ, δ)-PAC if, with a probability greater that $1 - \delta$, after using a finite number of samples, it returns a rank for each arm such that the i^{th} ranked cluster according to the returned ranking is a subset of $M^*_{i,\epsilon}$ for all $1 \leqslant i \leqslant c$. Formally, if $\sigma(a)$ is the rank of arm a returned by the algorithm after using a finite number of samples, we can define the empirical cluster i as

$$\hat{M}_i \coloneqq \{a : \kappa_{i-1} + 1 \leqslant \sigma(a) \leqslant \kappa_i\},$$

and we say the algorithm is (ϵ, δ)-PAC if

$$\mathbb{P}\left(\exists\, i \text{ such that } \hat{M}_i \not\subseteq M^*_{i,\epsilon}\right) \leqslant \delta. \tag{3.4}$$

3.5 Algorithm

Let $(\kappa_1, \ldots, \kappa_c = K)$ be the cluster boundaries. We describe here the `LUCBRank` algorithm using generic confidence intervals $\mathcal{I}_a = [L_a(t), U_a(t)]$, where t indexes rounds of the algorithm. Let $N_a(t)$ be the number of times arm a has been sampled up to round t, and $S_a(t)$ be the sum of rewards of arm a up to round t. Let $\hat{p}_a(t) = \frac{S_a(t)}{N_a(t)}$ be the corresponding empirical mean reward. Sort the arms in the decreasing order of their empirical mean rewards, and for $1 \leqslant i \leqslant c - 1$, let $J_i(t)$ denote the κ_i arms with the highest empirical mean rewards. Define

$$l^i_t \coloneqq \underset{a \in J_i(t)}{\arg\min}\ L_a(t), \qquad u^i_t \coloneqq \underset{a \notin J_i(t)}{\arg\max}\ U_a(t) \tag{3.5}$$

to be the two *critical* arms from $J_i(t)$ and $J^c_i(t)$ that are likely to be misclassified (see Fig. 3.1).

Algorithm 2 contains the pseudocode of `LUCBRank`, which is also depicted in Fig. 3.1. The algorithm maintains active cluster boundaries in the set C, where a cluster boundary i is active if the overlap of confidence intervals in J_i and J^c_i is not less than ϵ. In every round, it samples both the critical arms at every active cluster boundary (lines 11-15). At the end of every round, it checks if the critical arms at

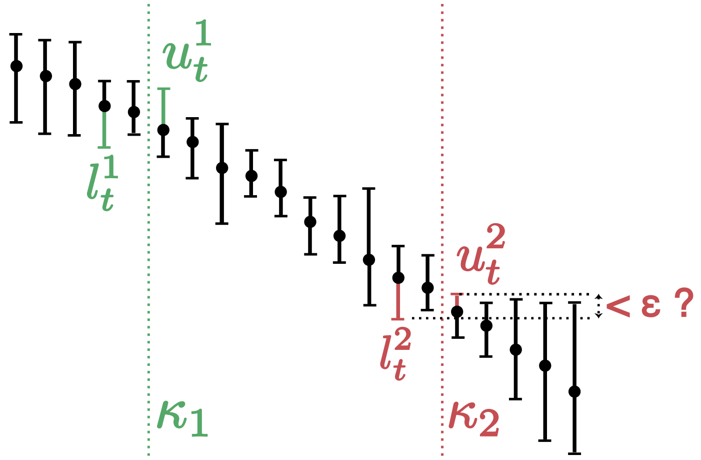


Figure 3.1: A visualization of LUCBRank on a bandit instance with $K = 20$ arms, $c = 3$ clusters, with boundaries at $\kappa_1 = 5, \kappa_2 = 15$. Also shown are the critical arms l_t^i, u_t^i pulled at each boundary. The algorithm stops sampling a boundary when the confidence interval overlap is less than ϵ.

any boundary are separated according to the tolerance criterion, and removes such boundaries from the active set (lines 21-25). For our experiments, we use KL-UCB [Garivier and Cappé, 2011] confidence intervals. For an exploration rate $\beta(t, \delta)$, the KL-UCB upper and lower confidence bounds for arm a are calculated as

$$\begin{aligned} U_a(t) &:= \max\{q \in [\hat{p}_a(t), 1] : N_a(t) d(\hat{p}_a(t), q) \leqslant \beta(t, \delta)\}, \\ L_a(t) &:= \min\{q \in [0, \hat{p}_a(t)] : N_a(t) d(\hat{p}_a(t), q) \leqslant \beta(t, \delta)\}. \end{aligned} \tag{3.6}$$

where $d(x, y)$ is the Kullback-Leibler divergence between two Bernoulli distributions, given by $d(x, y) = x \log \frac{x}{y} + (1 - x) \log \frac{1-x}{1-y}$.

LUCBRank can also be easily modified for pairwise-comparison queries: whenever the algorithm calls for drawing an arm i, duel arm i with another arm chosen uniformly at random.

Algorithm 2 `LUCBRank`

```
Input: $\epsilon > 0$, cluster boundaries $1 \leqslant \kappa_1, \ldots, \kappa_c = K$
$t \leftarrow 1$
$C \leftarrow \{1, \ldots, c-1\}$                    //active cluster boundaries

for $a = 1, \ldots, K$ do
    Sample item $a$, compute $U_a(1)$ and $L_a(1)$
end for

while $C \neq \varnothing$ do
    // Sample active cluster boundaries
    for $i \in C$ do
        Sample item $l_t^i$
        Sample item $u_t^i$
        (If pairwise comparing, compare item $l_t^i$ to a random other item, and
        compare item $u_t^i$ to a random other item. See Section 3.1.1)
    end for
    $t = t + 1$
    $\forall\, a \in [K]$ : Update reward-estimate $\hat{p}_a(t)$, number of samples $N_a(t)$, and
    confidence bounds $U_a(t), L_a(t)$ (see (3.6))
    $\forall\, i \in C$: Compute $l_t^i, u_t^i$ (see (3.5))

    // Eliminate unambiguous cluster boundaries
    for $i \in C$ do
        if $U_{u_t^i}(t) - L_{l_t^i}(t) < \epsilon$ then
            $C = C \setminus i$
        end if
    end for
end while

Return items sorted by their empirical mean rewards.
```

3.6 Analysis

We prove the accuracy of `LUCBRank` in Theorem 3.1, and give an upper bound on the sample complexity in Theorem 3.2. Our distribution-dependent lower bound for the sample complexity of any δ-PAC algorithm is stated in Theorem 3.4. All proofs can be found in the Appendix. Recall that $1 \leqslant \kappa_1 < \kappa_2 < \cdots < \kappa_{c-1} < \kappa_c = K$ are the cluster boundaries.

3.6.1 PAC Guarantee

Theorem 3.1 gives choices of $\beta(t, \delta)$ such that `LUCBRank` is correct with probability at least δ, in the sense defined by (3.4).

Theorem 3.1. *LUCBRank using* $\beta(t, \delta) = \log(\frac{k_1 K t^\alpha}{\delta}) + \log\log(\frac{k_1 K t^\alpha}{\delta})$ *with* $\alpha > 1$ *and* $k_1 > \left(\frac{c-1}{2}\right)^\alpha + \frac{2e}{\alpha-1} + \frac{4e}{(\alpha-1)^2}$*, is correct with probability* $1 - \delta$*.*

3.6.2 Sample Complexity

Our sample complexity results are stated in terms of Chernoff information [Cover and Thomas, 2012].

Chernoff Information: Consider two Bernoulli distributions $\mathcal{B}(x)$ and $\mathcal{B}(y)$, and let $d(x, y)$ denote the KL-divergence between these distributions. The Chernoff information $d^*(x, y)$ between these two Bernoulli distributions is defined by

$$d^*(x, y) := d(z^*, x) = d(z^*, y)$$

where z^* is the unique z such that $d(z, x) = d(z, y)$.

Next we introduce some notation. For an arm a, let $g(a)$ (read group of arm a) denote the index of the cluster that arm a belongs to. Formally,

$$g(a) := \min\{1 \leqslant i \leqslant c : p_a \leqslant p_{\kappa_i}\}. \tag{3.7}$$

Let $b_i \in [p_{\kappa_i}, p_{\kappa_i+1}], 1 \leqslant i \leqslant c-1$ be any points in the cluster boundary gaps, and

$b := (b_1, b_2, \dots, b_{c-1})$. Define

$$\Delta^*_b(a) := \begin{cases} d^*(p_a, b_1) & a \in \{1, \dots, \kappa_1\} \\ \min(d^*(p_a, b_{g(a)-1}), d^*(p_a, b_{g(a)}) & a \in \{\kappa_1 + 1, \dots, \kappa_{c-1}\} \\ d^*(p_a, b_{c-1}) & a \in \{\kappa_{c-1} + 1, \dots, K\} \end{cases} \tag{3.8}$$

to be the "distance" of each arm from the closest cluster boundary. Our upper bound on the sample complexity of `LUCBRank` is stated in Theorem 3.2, and contains the quantity $H^*_{\epsilon,b}$ where

$$H^*_{\epsilon,b} := \sum_{a \in \{1,\dots,K\}} \frac{1}{\max(\Delta^*_b(a), \epsilon^2/2)}. \tag{3.9}$$

Theorem 3.2. *Let* $b = (b_1, b_2, \dots, b_{c-1})$*, where* $b_i \in [p_{\kappa_i}, p_{\kappa_i+1}]$*. Let* $\epsilon > 0$*. Let* $\beta(t, \delta) = \log(\frac{k_1 K t^\alpha}{\delta}) + \log\log(\frac{k_1 K t^\alpha}{\delta})$ *with* $k_1 > \left(\frac{c-1}{2}\right)^\alpha + \frac{2e}{\alpha-1} + \frac{4e}{(\alpha-1)^2}$*. Let* τ *be the random number of samples taken by* `LUCBRank` *before termination. If* $\alpha > 1$,

$$\mathbb{P}\left(\tau \leqslant 2C_0(\alpha) H^*_{\epsilon,b} \log\left(\frac{k_1 K (2H^*_{\epsilon,b})^\alpha}{\delta}\right)\right) \geqslant 1 - \delta$$

where $C_0(\alpha)$ *is such that* $C_0(\alpha) \geqslant \left(1 + \frac{1}{e}\right)\left(\alpha \log(C_0(\alpha)) + 1 + \frac{\alpha}{e}\right)$.

3.6.3 Distribution-Dependent Lower Bound

In this section, we state our non-asymptotic lower bound on the expected number of samples needed by any δ-PAC algorithm to cluster and rank the arms into groups of sizes $(\kappa_1, \kappa_2 - \kappa_1, \dots, K - \kappa_{c-1})$. For simplicity, we focus on the case $\epsilon = 0$. The proof of the lower bound uses standard change of measure arguments [Kaufmann et al., 2015], which requires some continuity and well-separation assumptions. We state these next.

We consider the following class of bandit models where the clusters are unambiguously separated, i.e.

$$\mathcal{M}_\kappa = \{p = (p_1, \ldots, p_K) : p_i \in \mathcal{P}, p_{\kappa_i} > p_{\kappa_i+1}, 1 \leqslant i < c\}, \tag{3.10}$$

where $\mathcal{P}$ is a set that satisfies

$$\forall\, p, q \in \mathcal{P}^2, p \neq q \Rightarrow 0 < \mathrm{KL}(p, q) < +\infty.$$

We also assume the following:

Assumption 3.3. *For all* $p, q \in \mathcal{P}^2$ *such that* $p \neq q$, *for all* $\alpha > 0$,
there exists $q_1 \in \mathcal{P}$: *KL*$(p, q) <$ *KL*$(p, q_1) <$ *KL*$(p, q) + \alpha$ *and* $\mathbb{E}_{X\sim q_1}[X] > \mathbb{E}_{X\sim q}[X]$,
there exists $q_2 \in \mathcal{P}$: *KL*$(p, q) <$ *KL*$(p, q_2) <$ *KL*$(p, q) + \alpha$ *and* $\mathbb{E}_{X\sim q_2}[X] < \mathbb{E}_{X\sim q}[X]$.

To state our lower bound, we need to define for each arm a, another "distance" from the boundary, similar to (3.8). Define

$$\Delta_\kappa^{\mathrm{KL}}(a) := \begin{cases} \mathrm{KL}(p_a, p_{\kappa_1+1}) & a \in \{1, \ldots, \kappa_1\} \\ \min(\mathrm{KL}(p_a, p_{\kappa_{g(a)-1}}), \mathrm{KL}(p_a, p_{\kappa_{g(a)}+1}) & a \in \{\kappa_1 + 1, \ldots, \kappa_{c-1}\} \\ \mathrm{KL}(p_a, p_{\kappa_{c-1}}) & a \in \{\kappa_{c-1} + 1, \ldots, K\}, \end{cases} \tag{3.11}$$

where $g(a)$ defined in (3.7) is the cluster that arm a belongs to. We highlight the differences from (3.8). First, the Chernoff information in (3.8) is replaced with KL-divergence in (3.11), and second, the distance is measured with the closest arm in either adjacent cluster here, as opposed to a point in the gap between the clusters in (3.8).

Our lower bound involves the quantity

$$\sum_{a \in 1,\ldots,K} \frac{1}{\Delta_\kappa^{\mathrm{KL}}(a)} \tag{3.12}$$

and is as follows:

Theorem 3.4. *Let* $p \in \mathcal{M}_\kappa$*, and assume that* $\mathcal{P}$ *satisfies Assumption 3.3; any coarse ranking algorithm that is* δ*-PAC on* $\mathcal{M}_\kappa$ *satisfies, for* $\delta \leqslant 0.15$*,*

$$\mathbb{E}_p[\tau] \geqslant \left[\sum_{a \in 1,\dots,K} \frac{1}{\Delta_\kappa^{KL}(a)}\right] \log\left(\frac{1}{2.4\delta}\right)$$

3.6.4 Remarks

- The tightest high-probability upper bound is obtained by setting b equal to $\underset{b:b_i \in [p_{\kappa_i}, p_{\kappa_i+1}]}{\arg\min} \quad H^*_{\epsilon,b}$ in Theorem 3.2.
- Although stated for Bernoulli distributions, the results in this chapter can easily be extended to rewards in the exponential family [Garivier and Cappé, 2011] by using the appropriate d function.

3.7 Experiments

3.7.1 Ranking from Direct Feedback

We first compare `LUCBRank` with uniform sampling and the Active Ranking (AR) algorithm [Heckel et al., 2016]. AR is an adaptation of the successive elimination approach to solve the coarse ranking problem. It maintains a set of unranked items and samples every item in this set, removing an item from the set when it is confident of the cluster the item belongs to. Although developed for pairwise comparison feedback, AR can easily be adapted to the direct-feedback setting.

We look at the bandit instance B with $K = 15$ arms whose rewards are Bernoulli distributed with means $(p_1 = \frac{1}{2}; p_a = \frac{1}{2} - \frac{a}{40}$ for $a = 2, 3, \dots, K)$. This problem has been studied in the literature in the context of finding the best-arm [Bubeck et al., 2013]. We consider the problem of finding the top-3 and the bottom-3 arms, which corresponds to $\kappa_1 = 3, \kappa_2 = 12$.

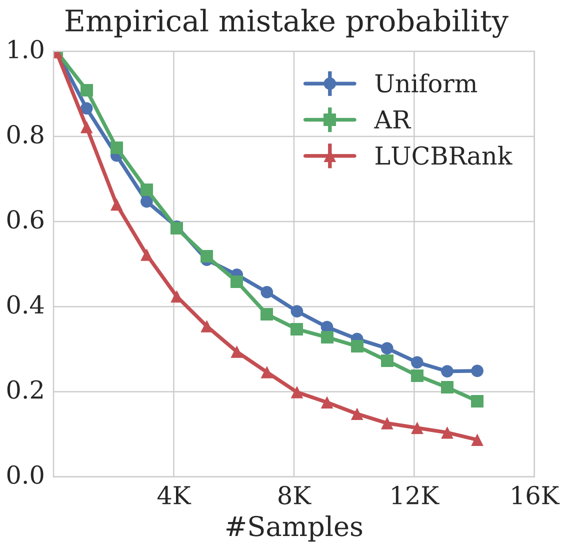
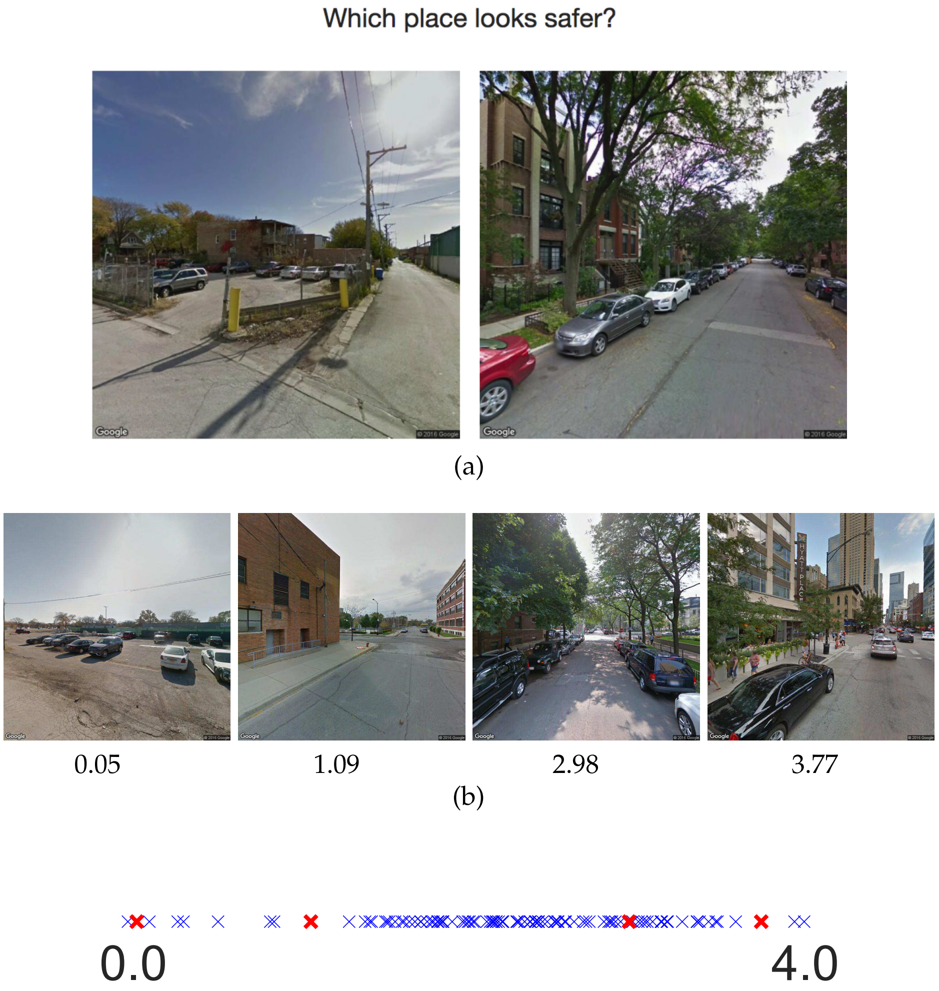


Figure 3.2: Exp 1 (description in text)

In Fig. 3.2, we record the probability (averaged over 1000 simulations) that the empirical clusters returned by the algorithm do not match the true clusters. We set $\delta = 0.1$ for both LUCBRank and AR, and $\epsilon = 0$ in LUCBRank to have a fair comparison with AR. We see that the mistake probability drops faster for LUCBRank than for AR.

3.7.2 Ranking from Pairwise Comparisons

To measure the performance of our algorithm on real-world data, we selected $K = 100$ Google street view images in Chicago, and collected 6000 pairwise responses on MTurk using NEXT [Jamieson et al., 2015a], where we asked users to choose the safer-looking image out of two images. This experiment is similar to the Place Pulse project [Naik et al., 2014], where the objective is to assess how the appearance of a neighborhood affects its perception of safety. Fig. 3.3(a) shows a sample query from our experiment. We estimated the safety scores of these street view images from the user-responses by fitting a Bradley-Terry-Luce (BTL) model [Bradley and Terry, 1952] using maximum likelihood estimation, and used this as the ground truth to generate noisy comparisons. Given two items i and j with scores θ_i and



(a)

(b)

(c)

Figure 3.3: (a) A sample query on NEXT. (b) Four sample images and their estimated BTL scores beneath. (c) Scatter plot of all the BTL scores, with the sample image markers highlighted.

θ_j, the BTL model estimates the probability that item i is preferred to item j as $\mathbb{P}(i > j) = \frac{e^{\theta_i}}{e^{\theta_i}+e^{\theta_j}}$. Fig. 3.3(b) shows 4 images overlayed with their estimated BTL scores (where the lowest score was set to 0), and Fig. 3.3(c) shows a scatter plot of the scores of all 100 images.

We first study the performance of adaptive methods with the goal of finding a

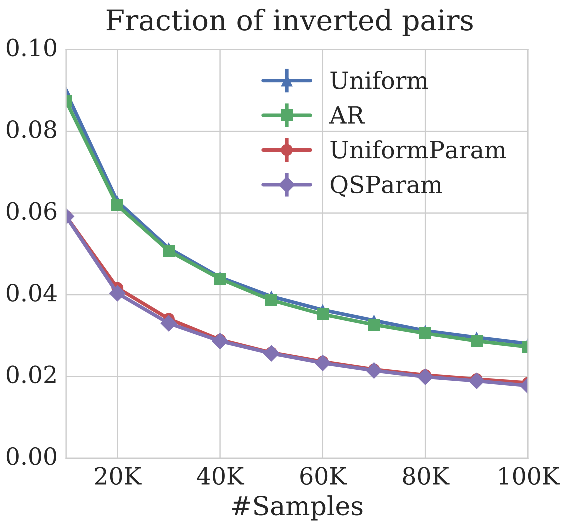


Figure 3.4: The futility of adaptive methods if the goal is to obtain a complete ranking. We compare uniform sampling with Active Ranking (both use non-parametric rank aggregation), and uniform sampling with quicksort (where both use parametric rank aggregation). We see that the Kendall tau distance of adaptive methods is no lower than those of their non-adaptive counterparts.

complete ranking, and observe that adaptive methods offer no advantages when items means are close to each other as they are in this dataset. Oblivious of the generative model, a lower bound (ignoring constants and log factors) on the number of samples required to sort the items by their Borda scores is given by $\sum 1/\Delta_i^2$ [Jamieson et al., 2015b], where the Δ_is are gaps between consecutive sorted Borda scores. For the dataset considered in this experiment, $\sum 1/\Delta_i^2 = 322$ million! We verify the futility of adaptive methods in Fig. 3.4, where we compare the performance of parametric as well as non-parametric adaptive methods in the literature (we describe these methods shortly) to their non-adaptive counterparts, with a goal of finding a complete ranking of the images. In the parametric algorithms (UniformParam and QSParam), we find MLE estimates of the BTL scores that best fit the pairwise responses. In the non-parametric algorithms (Uniform and AR), we estimate the scores using empirical probabilities in Eq. (3.2). In Fig. 3.4, we plot the fraction of pairs that are inverted in the empirical ranking compared to the true ranking, and see no benefits for adaptive methods. We do see gains from adaptivity in the coarse formulation (Fig. 3.5), as we explain next.

LUCBRank can be used in the pairwise comparison setting using Borda reduction, as described in Section 3.1.1. The adaptive methods in literature we compare to are AR (as in the previous section), and Quicksort (QS) [Ailon et al., 2008, Maystre and Grossglauser, 2017]. The Quicksort algorithm works exactly like its non-noisy counterpart: it compares a randomly chosen pivot to all elements, and divides the elements into two subsets - elements preferred to the pivot, and elements the pivot was preferred over. The algorithm then recurses into these two subsets. In this experiment, we stop the quicksort algorithm early as soon as all the subsets are inside the user-specified clusters. Continuing the algorithm further won't change the items in any cluster. This reduces the sample complexity of Quicksort.

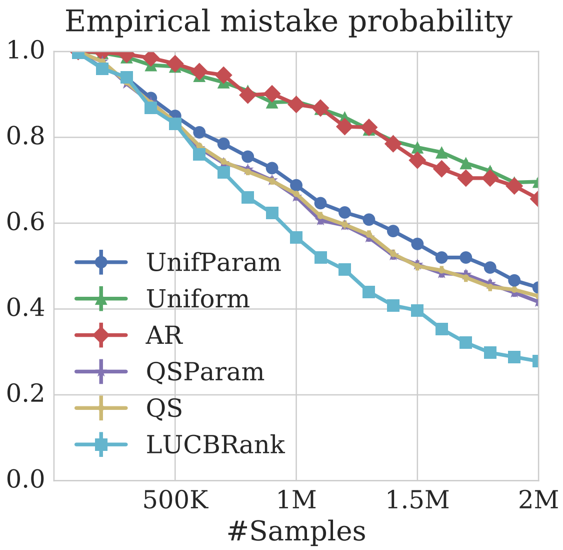


Figure 3.5: Probability of error in identifying the clusters: LUCBRank does better than parametric versions of other active algorithms.

We consider the problem of clustering the images into pentiles ($\kappa_i = 20\,i,\ 1 \leqslant i \leqslant 5$). We set $\delta = 0.1$ for both LUCBRank and AR, and $\epsilon = 0$ in LUCBRank to ensure a fair comparison with AR. In Fig. 3.5, we record the probability (averaged over 600 simulations) that the empirical pentiles returned by the algorithm do not match the true pentiles. We find that LUCBRank has a lower mistake probability than even the parametric version of Quicksort, which assumes knowledge of the BTL model. As an aside, note that when the items are close as in this experiment, the parametric versions of Uniform and Quicksort perform similarly, and the active nature of Quicksort offers no significant advantage.

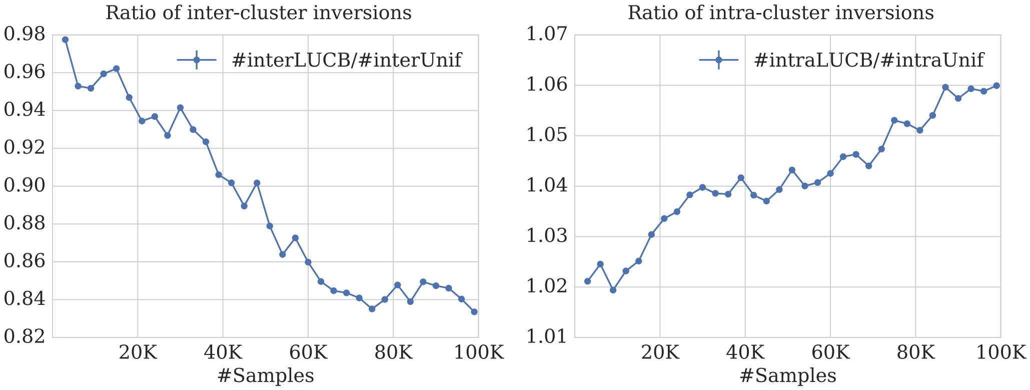


Figure 3.6: (a) The ratio of inter-cluster inversions of LUCBRank and Uniform. (b) The ratio of intra-cluster inversions of LUCBRank and Uniform. LUCBRank focuses on minimizing inter-cluster inversions at the cost of intra-cluster inversions.

In Fig. 3.6(a) and (b) we plot the ratio of inter-cluster and intra-cluster inversions respectively of LUCBRank and Uniform. An inter-cluster pair is a pair of items that are in different clusters in the true ranking, while an intra-cluster pair is a pair of items from the same cluster. We see the that ratio of inter-cluster inversions goes down in Fig. 3.6(a), because that is the metric LUCBRank focuses on. LUCBRank does not expend effort on refining its estimate of an item's rank once its cluster has been found, and hence pays a price in the form of intra-cluster inversions (Fig. 3.6(b)).

3.8 Conclusion

The coarse ranking setting is motivated from real-world problems where humans rate items. These problems have high noise and are hard, and a complete ranking is not feasible; fortunately, it is often also not necessary. We propose a practical online algorithm for solving it, `LUCBRank`, and prove distribution-dependent upper and lower bounds on its sample complexity. We evaluate its performance on crowdsourced data gathered using MTurk, and observe that it performs better than existing algorithms in the literature.

We leave open several questions. First, our upper bound is stated in terms of Chernoff information between distributions, while our lower bound is in terms of KL-divergences, and there is a gap between the two. Second, the cluster boundaries need to be user-specified in our current setting. If the gap between the nearest items in adjacent clusters is small, this can adversely affect the sample complexity. Although this is partially addressed through the error-tolerance ϵ, an attractive algorithm would be one which auto-tunes the positions of the cluster boundaries at the widest gaps, subject to user-specified constraints.

To the best of our knowledge, this chapter presents the first bandit UCB algorithm for ranking.

3.9 Appendix

3.9.1 PAC Guarantee

We'll use the following lemma [Kaufmann and Kalyanakrishnan, 2013] which bounds the probability of 'bad' events in round t.

Lemma 3.5. *Let* $U_a(t)$ *and* $L_a(t)$ *be the confidence bounds defined in Eq.* (3.6). *For any algorithm and arm* a,

$$\mathbb{P}(U_a(t) < p_a) = \mathbb{P}(L_a(t) > p_a) \leqslant e(\beta(t,\delta)\log t + 1)\exp(-\beta(t,\delta))$$

We shall also need the following technical lemma, which we'll use to upper bound the probability of any bad event.

Lemma 3.6. *If* $\beta(t,\delta) = \log(\frac{k_1 K t^\alpha}{\delta}) + \log\log(\frac{k_1 K t^\alpha}{\delta})$,

$$\sum_{t=1}^{\infty} (\beta(t,\delta)\log t + 1)\exp(-\beta(t,\delta)) \leqslant \frac{\delta}{k_1 K}\left(\frac{2}{(\alpha-1)^2} + \frac{1}{(\alpha-1)}\right)$$

Proof. Let us consider

$$\begin{aligned}
\beta(t,\delta)(\log t)e^{-\beta(t,\delta)} &= \left(\log\left(\tfrac{k_1 K t^\alpha}{\delta}\right) + \log\log\left(\tfrac{k_1 K t^\alpha}{\delta}\right)\right)(\log t)\left(\frac{\delta}{k_1 K t^\alpha}\cdot\frac{1}{\log\frac{k_1 K t^\alpha}{\delta}}\right) \\
&\leqslant 2\log\left(\frac{k_1 K t^\alpha}{\delta}\right)\cdot\log t\cdot\left(\frac{\delta}{k_1 K t^\alpha}\cdot\frac{1}{\log\frac{k_1 K t^\alpha}{\delta}}\right) \\
&= 2\log t\cdot\frac{\delta}{k_1 K t^\alpha}
\end{aligned}$$

Hence

$$\begin{aligned}
\sum_{t=1}^{\infty} (\beta(t,\delta)\log t + 1)\exp(-\beta(t,\delta)) &\leqslant \sum_{t=1}^{\infty}\left(2\log t\cdot\frac{\delta}{k_1 K t^\alpha} + \frac{\delta}{k_1 K t^\alpha}\right) \\
&\leqslant \frac{\delta}{k_1 K}\left(\frac{2}{(\alpha-1)^2} + \frac{1}{(\alpha-1)}\right)
\end{aligned}$$

□

3.9.1.1 Proof of Theorem 3.1

Theorem. *LUCBRank using* $\beta(t,\delta) = \log(\frac{k_1 K t^\alpha}{\delta}) + \log\log(\frac{k_1 K t^\alpha}{\delta})$ *with* $\alpha > 1$ *and* $k_1 > 1 + \frac{2e}{\alpha-1} + \frac{4e}{(\alpha-1)^2}$, *is correct with probability* $1-\delta$.

Proof. Consider the event

$$W = \bigcap_{t\in\mathbb{N}} \bigcap_{a\in\{1,\ldots,K\}} ((U_a(t) > p_a) \cap (L_a(t) < p_a))$$

where all arms are well-behaved i.e. their true means are inside their confidence intervals. We show that LUCBRank is correct on the event W.

Assume LUCBRank fails, which means that when it terminates, there exists a cluster i, such that arm a belongs to cluster i in the returned ranking, and $a \in M^{*,c}_{\epsilon,i}$; that is, either 1) $p_a > p_{\kappa_{i-1}+1} + \epsilon$ or 2) $p_a < p_{\kappa_i} - \epsilon$.

Consider the first case: $p_a > p_{\kappa_{i-1}+1} + \epsilon$. Consequently, there exists arm b such that $p_b \leqslant p_{\kappa_{i-1}+1}$, and $\tau(b) \leqslant \kappa_{i-1}$ in the returned ranking. Since the algorithm stopped and boundary $i-1$ was removed from the set of active boundaries C, it must be the case that $U_a(t) - L_b(t) < \epsilon$ upon stopping. Hence, the following holds:

$$
\begin{aligned}
&\bigcup_{t\in\mathbb{N}} (\exists\, a, b : p_a > p_{\kappa_{i-1}+1} + \epsilon, p_b \leqslant p_{\kappa_{i-1}+1}, U_a(t) - L_b(t) < \epsilon) \\
\subseteq &\bigcup_{t\in\mathbb{N}} (\exists\, a, b : (U_a(t) < p_b + \epsilon < p_a) \cup (L_b(t) > p_b)) \\
\subseteq &\bigcup_{t\in\mathbb{N}} \bigcup_{a\in\{1,\ldots,K\}} (U_a(t) < p_a) \bigcup_{b\in\{1,\ldots,K\}} (L_b(t) > p_b) \subseteq W^c
\end{aligned}
$$

Consider the second case: $p_a < p_{\kappa_i} - \epsilon$. Consequently, there exists an arm b such that $p_b \geqslant p_{\kappa_i}$, and $\tau(b) > \kappa_i$ in the returned ranking. Since the algorithm stopped and boundary i was removed from the set of active boundaries C, it must be the case that $U_b(t) - L_a(t) < \epsilon$ upon stopping. Hence, the following holds:

$$
\begin{aligned}
&\bigcup_{t\in\mathbb{N}} (\exists\, a, b : p_a < p_{\kappa_i} - \epsilon, p_b \geqslant p_{\kappa_i}, U_b(t) - L_a(t) < \epsilon) \\
\subseteq &\bigcup_{t\in\mathbb{N}} (\exists\, a, b : (U_b(t) < p_b) \cup (L_a(t) > p_b - \epsilon > p_a)) \\
\subseteq &\bigcup_{t\in\mathbb{N}} \bigcup_{b\in\{1,\ldots,K\}} (U_b(t) < p_b) \bigcup_{a\in\{1,\ldots,K\}} (L_a(t) > p_a) \subseteq W^c
\end{aligned}
$$

Hence

$$
\begin{aligned}
\mathbb{P}(\texttt{LUCBRank}\text{ fails}) &\leqslant \mathbb{P}(W^c) \\
&\leqslant 2eK\sum_{t=1}^{\infty}(\beta(t,\delta)\log t+1)\exp(-\beta(t,\delta)) && \text{(by Lemma 3.5)} \\
&\leqslant \frac{\delta}{k_1}\left(\frac{4e}{(\alpha-1)^2}+\frac{2e}{(\alpha-1)}\right) && \text{(by Lemma 3.6)} \\
&\leqslant \delta && \text{(by the constraint on } k_1\text{)}
\end{aligned}
$$

□

3.9.2 Sample Complexity

We define the event W_t which says that all arms are well-behaved in round t i.e. their true means are contained inside their confidence intervals.

$$W_t = \bigcap_{a\in\{1,2,\ldots,K\}} ((U_a(t) > p_a)\cap(L_a(t) < p_a))$$

Note that the event W defined earlier is $W = \cup_{t\in\mathbb{N}} W_t$.

Proposition 3.7 gives a sufficient condition for stopping.

Proposition 3.7. *Let* $b_i \in [p_{\kappa_i}, p_{\kappa_i+1}]$. *If* $U_{u_t^i} - L_{l_t^i} > \epsilon$ *and* W_t *holds, then either* $k = l_t^i$ *or* $k = u_t^i$ *satisfies*

$$b_i \in \mathcal{I}_k(t) \textit{ and } \tilde{\beta}_k(t) > \frac{\epsilon}{2},$$

where we define $\tilde{\beta}_a(t) = \sqrt{\frac{\beta(t,\delta)}{2N_a(t)}}$

Proof. Our W_t condition is stronger than that required in the Proposition 1 in Kaufmann and Kalyanakrishnan [2013], and hence their proof applies. □

Lemma 3.8 is another concentration result that will be used in our sample complexity guarantee.

Lemma 3.8. *Let* $\mathsf{T} \geqslant 1$ *be an integer, and* $1 \leqslant i \leqslant (c-1)$ *be any cluster boundary. Let* $\delta > 0, \gamma > 0$ *and* $x \in]0,1[$ *be such that* $p_a \neq x$. *Then*

$$\sum_{t=1}^{T} \mathbb{P}\left(a = u_i^t \vee a = l_i^t, N_a(t) > \left\lceil \frac{\gamma}{d^*(p_a,x)} \right\rceil, N_a(t) d(\hat{p}_a(t),x) \leqslant \gamma\right) \leqslant \frac{\exp(-\gamma)}{d^*(p_a,x)}$$

We prove the following lemma, which states that the Chernoff information increases as the second distribution moves away from the first.

Lemma 3.9. *If* $x < y < y'$ *or* $x > y > y'$, $d^*(x,y) \leqslant d^*(x,y')$

Proof. We shall prove the statement for the case $x < y < y'$. The proof for $x > y > y'$ is analogous.
Let z^* be the unique z such that $d(z^*,x) = d(z^*,y)$. Since $z^* < y < y'$, $d(z^*,y') \geqslant d(z^*,y) = d(z^*,x)$. Hence, there exists $z^{*'} \geqslant z^*$ such that $d^*(x,y) = d(z^*,x) \leqslant d(z^{*'},x) = d(z^{*'},y') = d^*(x,y')$. □

Lemma 3.10. *Let* x^* *be the solution of the equation:*

$$x = \frac{1}{\gamma}\left(\log\frac{x^\alpha}{\eta} + \log\log\frac{x^\alpha}{\eta}\right)$$

Then if $\gamma < 1$ *and* $\eta < 1/e^e$,

$$\frac{1}{\gamma}\log\left(\frac{1}{\eta\gamma^\alpha}\right) \leqslant x^* \leqslant \frac{C_0}{\gamma}\log\left(\frac{1}{\eta\gamma^\alpha}\right)$$

where C_0 *is such that* $C_0 \geqslant \left(1+\frac{1}{e}\right)\left(\alpha\log C_0 + 1 + \frac{\alpha}{e}\right)$.

Proof. x^* is upper bounded by any x such that $\frac{1}{\gamma}\left(\log\frac{x^\alpha}{\eta} + \log\log\frac{x^\alpha}{\eta}\right) \leqslant x$. We look for x^* of the form $\frac{C_0}{\gamma}\log\left(\frac{1}{\eta\gamma^\alpha}\right)$.

$$\begin{aligned}
\frac{1}{\gamma}\left(\log\frac{x^\alpha}{\eta}+\log\log\frac{x^\alpha}{\eta}\right) &\leqslant \frac{1}{\gamma}\left(1+\frac{1}{e}\right)\log\left(\frac{x^\alpha}{\eta}\right)\\
&= \frac{1}{\gamma}\left(1+\frac{1}{e}\right)\left(\alpha\log C_0+\log\frac{1}{\eta\gamma^\alpha}+\alpha\log\log\frac{1}{\eta\gamma^\alpha}\right)\\
&\leqslant \frac{1}{\gamma}\left(1+\frac{1}{e}\right)\left(\alpha\log C_0+\left(1+\frac{\alpha}{e}\right)\log\frac{1}{\eta\gamma^\alpha}\right)\\
&\leqslant \frac{1}{\gamma}\left(1+\frac{1}{e}\right)\left(\alpha\log C_0+1+\frac{\alpha}{e}\right)\log\frac{1}{\eta\gamma^\alpha}
\end{aligned}$$

where the first and second inequalities hold because $\log x \leqslant \frac{x}{e}$, and the last inequality holds because $\frac{1}{\eta\gamma^\alpha} > e$. Choosing C_0 such that

$$C_0 \geqslant \left(1+\frac{1}{e}\right)\left(\alpha\log C_0+1+\frac{\alpha}{e}\right)$$

gives us our upper bound.

To prove the lower bound, consider the series defined by

$$\begin{aligned}
x_0 &= 1\\
x_{n+1} &= \frac{1}{\gamma}\left(\log\frac{x_n^\alpha}{\eta}+\log\log\frac{x_n^\alpha}{\eta}\right)
\end{aligned}$$

First note that since $\gamma < 1$ and $\eta < 1/e^e$, the sequence is increasing. Second, note that the sequence converges to x^*. Hence

$$\begin{aligned}
x^* \geqslant x_2 &= \frac{1}{\gamma}\left[\log\left(\frac{1}{\eta\gamma^\alpha}\left(\log\frac{1}{\eta}+\log\log\frac{1}{\eta}\right)^\alpha\right)+\log\log\left(\frac{1}{\eta\gamma^\alpha}\left(\log\frac{1}{\eta}+\log\log\frac{1}{\eta}\right)^\alpha\right)\right]\\
&= \frac{1}{\gamma}\left[\log\frac{1}{\eta\gamma^\alpha}+\alpha\log\left(\log\frac{1}{\eta}+\log\log\frac{1}{\eta}\right)+\log\log\frac{1}{\eta\gamma^\alpha}+\alpha\log\log\left(\log\frac{1}{\eta}+\log\log\frac{1}{\eta}\right)\right]\\
&\geqslant \frac{1}{\gamma}\log\frac{1}{\eta\gamma^\alpha}
\end{aligned}$$

since $\eta < 1/e^e$. □

Corollary 3.11. *Let* $\gamma = \frac{1}{2H^*_{\epsilon,b}}, \eta = \frac{\delta}{k_1 K}$. *Then applying Lemma 3.10 gives*

$$2H^*_{\epsilon,b} \log\left(\frac{k_1 K (2H^*_{\epsilon,b})^\alpha}{\delta}\right) \leqslant S^*_1 \leqslant 2C_0(\alpha) H^*_{\epsilon,b} \log\left(\frac{k_1 K (2H^*_{\epsilon,b})^\alpha}{\delta}\right)$$

3.9.3 Proof of Theorem 3.2

Theorem. *Let* $b = (b_1, b_2, \ldots, b_{c-1})$, *where* $b_i \in [p_{\kappa_i}, p_{\kappa_i+1}]$. *Let* $\epsilon > 0$. *Let* $\beta(t,\delta) = \log(\frac{k_1 K t^\alpha}{\delta}) + \log\log(\frac{k_1 K t^\alpha}{\delta})$ *with* $k_1 > 1 + \frac{2e}{\alpha-1} + \frac{4e}{(\alpha-1)^2}$. *Let* τ *be the random number of samples taken by* `LUCBRank` *before termination. If* $\alpha > 1$,

$$\mathbb{P}\left(\tau \leqslant 2C_0(\alpha) H^*_{\epsilon,b} \log\left(\frac{k_1 K (2H^*_{\epsilon,b})^\alpha}{\delta}\right)\right) \geqslant 1 - \delta$$

where $C_0(\alpha)$ *is such that* $C_0(\alpha) \geqslant \left(1 + \frac{1}{e}\right)\left(\alpha \log(C_0(\alpha)) + 1 + \frac{\alpha}{e}\right)$.

Proof. The `LUCBRank` algorithm proceeds in rounds. In a round, it samples the two arms on opposite sides of an active boundary whose confidence intervals overlap the most. A boundary is active as long as this overlap is less than ϵ. Thus, the number of samples up to round T is

$$\begin{aligned}
\#\text{samples}(T) &\leqslant 2\sum_{t=1}^{T}\sum_{i=1}^{c-1} \mathbb{1}_{(U_{u^i_t} - L_{l^i_t} > \epsilon)} \\
&= 2\sum_{t=1}^{T}\sum_{i=1}^{c-1} \mathbb{1}_{(U_{u^i_t} - L_{l^i_t} > \epsilon)} (\mathbb{1}_{W_t} + \mathbb{1}_{W^c_t}) \\
&\leqslant 2\sum_{t=1}^{T}\sum_{i=1}^{c-1} \mathbb{1}_{(U_{u^i_t} - L_{l^i_t} > \epsilon)} \mathbb{1}_{W_t} + 2\sum_{t=1}^{T}\sum_{i=1}^{c-1} \mathbb{1}_{W^c_t} \\
&\leqslant 2\sum_{t=1}^{T}\sum_{i=1}^{c-1} \sum_{a\in\{1,2,\ldots,K\}} \mathbb{1}_{(a=l^i_t)\vee(a=u^i_t)} \mathbb{1}_{(b_i \in \mathcal{I}_a(t))} \mathbb{1}_{(\tilde{\beta}_a(t) > \frac{\epsilon}{2})} + 2\sum_{t=1}^{T}\sum_{i=1}^{c-1} \mathbb{1}_{W^c_t} \\
&\quad \text{(by Proposition 3.7)}
\end{aligned}$$

We now split the first sum into two depending on whether an arm a belongs to the set $\mathcal{A}_\epsilon = \{a \in \{1, 2, ..., K\} : \Delta_b^* < \epsilon^2/2\}$.

$$\begin{aligned}
\#\text{samples}(T) &\leqslant 2 \sum_{a \in \mathcal{A}_\epsilon} \sum_{t=1}^{T} \sum_{i=1}^{c-1} \mathbb{1}_{(a=l_t^i) \vee (a=u_t^i)} \mathbb{1}_{\left(N_a(t) < \frac{\beta(t,\delta)}{\epsilon^2/2}\right)} + \\
&\quad 2 \sum_{a \in \mathcal{A}_\epsilon^c} \sum_{t=1}^{T} \sum_{i=1}^{c-1} \mathbb{1}_{(a=l_t^i) \vee (a=u_t^i)} \mathbb{1}_{(b_i \in \mathcal{I}_a(t))} + 2 \sum_{t=1}^{T} \sum_{i=1}^{c-1} \mathbb{1}_{\mathcal{W}_t^c} \\
&\leqslant 2 \sum_{a \in \mathcal{A}_\epsilon} \frac{\beta(T,\delta)}{\epsilon^2/2} + 2 \sum_{a \in \mathcal{A}_\epsilon^c} \sum_{t=1}^{T} \sum_{i=1}^{c-1} \mathbb{1}_{(a=l_t^i) \vee (a=u_t^i)} \mathbb{1}_{\left(N_a(t) \leqslant \left\lceil \frac{\beta(T,\delta)}{\Delta_b^*(a)} \right\rceil\right)} + \\
&\quad \underbrace{2 \sum_{a \in \mathcal{A}_\epsilon^c} \sum_{t=1}^{T} \sum_{i=1}^{c-1} \mathbb{1}_{(a=l_t^i) \vee (a=u_t^i)} \mathbb{1}_{\left(N_a(t) > \left\lceil \frac{\beta(T,\delta)}{\Delta_b^*(a)} \right\rceil\right)} + 2 \sum_{t=1}^{T} \sum_{i=1}^{c-1} \mathbb{1}_{\mathcal{W}_t^c}}_{R_T} \\
&= 2H_{\epsilon,b}^* \, \beta(T,\delta) + R_T
\end{aligned}$$

where

$$R_T = 2 \sum_{a \in \mathcal{A}_\epsilon^c} \sum_{t=1}^{T} \sum_{i=1}^{c-1} \mathbb{1}_{(a=l_t^i) \vee (a=u_t^i)} \mathbb{1}_{\left(N_a(t) > \left\lceil \frac{\beta(T,\delta)}{\Delta_b^*(a)} \right\rceil\right)} \mathbb{1}_{(b_i \in \mathcal{I}_a(t))} + 2 \sum_{t=1}^{T} \sum_{i=1}^{c-1} \mathbb{1}_{\mathcal{W}_t^c}$$

If we define $S_1^* = \min\{x : 2H_{\epsilon,b}^* \, \beta(x,\delta) < x\}$, then we get that for $S > S_1^*$, the algorithm must have stopped before S samples on the event $(R_T = 0)$. Denoting the total number of samples used by the algorithm by τ, we have that, for any $S > S_1^*$, $\mathbb{P}(\tau > S) \leqslant \mathbb{P}(R_T \neq 0)$.

$$\begin{aligned}
&\mathbb{P}(\tau > S) \leqslant \mathbb{P}(R_T \neq 0) \\
&\leqslant \mathbb{P}\left(\exists\, a \in \mathcal{A}_\epsilon^c, t \leqslant T, 1 \leqslant i \leqslant (c-1) : a = l_t^i \vee a = r_t^i, N_a(t) > \left\lceil \frac{\beta(T,\delta)}{\Delta_b^*(a)} \right\rceil, b_i \in \mathcal{I}_a(t)\right) + \mathbb{P}(\mathcal{W}^c) \\
&\leqslant \mathbb{P}\left(\exists\, a \in \mathcal{A}_\epsilon^c, t \leqslant T, 1 \leqslant i \leqslant (c-1) : a = l_t^i \vee a = r_t^i, N_a(t) > \left\lceil \frac{\beta(T,\delta)}{d^*(p_a, b_i)} \right\rceil, b_i \in \mathcal{I}_a(t)\right) + \mathbb{P}(\mathcal{W}^c)
\end{aligned} \tag{3.13}$$

where the final inequality follows because $\Delta^*_b(a) \leqslant d^*(p_a, b_i)\, \forall\, 1 \leqslant i \leqslant c-1$ (by Lemma 3.9).

Let us look at the first term:

$$
\begin{aligned}
&\mathbb{P}\left(\exists\, a \in \mathcal{A}^c_\epsilon, t \leqslant T, 1 \leqslant i \leqslant (c-1) : a = l^i_t \vee a = r^i_t, N_a(t) > \left\lceil \frac{\beta(T,\delta)}{d^*(p_a, b_i)} \right\rceil, b_i \in \mathcal{I}_a(t)\right) \\
&\leqslant \sum_{a \in \mathcal{A}^c_\epsilon} \sum_{i=1}^{c-1} \sum_{t=1}^{T} \mathbb{P}\left(a = l^i_t \vee a = r^i_t, N_a(t) > \left\lceil \frac{\beta(T,\delta)}{d^*(p_a, b_i)} \right\rceil, b_i \in \mathcal{I}_a(t)\right) \\
&\leqslant \sum_{a \in \mathcal{A}^c_\epsilon} \sum_{i=1}^{c-1} \frac{\exp(-\beta(T,\delta))}{d^*(p_a, b_i)} \quad \text{(by Lemma 3.8)} \\
&\leqslant (c-1)\exp(-\beta(T,\delta)) \sum_{a \in \mathcal{A}^c_\epsilon} \frac{1}{d^*(p_a, b_i)} \\
&\leqslant (c-1) H^*_{\epsilon,b} \exp(-\beta(T,\delta)) \\
&\leqslant (c-1) H^*_{\epsilon,b} \exp\left(-\beta\left(\tfrac{S^*_1}{c-1}, \delta\right)\right) \qquad (\text{if } \tau > S^*_1, T > \tfrac{S^*_1}{c-1})) \\
&= (c-1) H^*_{\epsilon,b} \cdot \frac{\delta (c-1)^\alpha}{k_1 K S^{*,\alpha}_1} \frac{1}{\log \frac{k_1 K S^{*,\alpha}_1}{\delta (c-1)^\alpha}} \\
&\leqslant (c-1)^{\alpha+1} H^*_{\epsilon,b} \cdot \frac{\delta}{k_1 K \left(2 H^*_{\epsilon,b} \log\left(\frac{k_1 K (2 H^*_{\epsilon,b})^\alpha}{\delta}\right)\right)^\alpha} \frac{1}{\log \frac{k_1 K S^{*,\alpha}_1}{\delta (c-1)^\alpha}} \qquad \text{(by the lower bound in Corollary} \\
&\leqslant \frac{\delta}{k_1} \cdot \left(\frac{c-1}{2}\right)^\alpha \qquad (\text{since } (c-1) \leqslant K \text{ and } \alpha > 1)
\end{aligned}
$$

For the second term, note that

$$
\begin{aligned}
\mathbb{P}(W^c) &\leqslant 2eK \sum_{t=1}^{\infty} (\beta(t,\delta)\log t + 1) \exp(-\beta(t,\delta)) \quad \text{(by Lemma 3.5)} \\
&\leqslant \frac{\delta}{k_1} \left(\frac{4e}{(\alpha-1)^2} + \frac{2e}{(\alpha-1)}\right) \qquad \text{(by Lemma 3.6)}
\end{aligned}
$$

Substituting in Eq. (3.13), we get that for $S > S^*_1$,

$$
\mathbb{P}(\tau > S) \leqslant \frac{\delta}{k_1} \left(\left(\frac{c-1}{2}\right)^\alpha + \frac{4e}{(\alpha-1)^2} + \frac{2e}{(\alpha-1)}\right) \leqslant \delta
$$

by the choice of k_1. □

3.9.4 Lower Bound

The proof uses standard change of measure arguments used to prove lower bounds. For bandit problems, this is succinctly expressed through Lemma 1 in Kaufmann et al. [2015] that we restate here for completeness.

Lemma 3.12. *Let* p *and* p' *be two bandit models with* K *arms such that for all* a*, the distributions* p_a *and* p'_a *are mutually absolutely continuous. Let* σ *be a stopping time with respect to* $(\mathcal{F}_t)$ *and let* $A \in \mathcal{F}_\sigma$*. Then*

$$\sum_{a=1}^{K} \mathbb{E}_p[N_a(\sigma)] KL(p_a, p'_a) \geqslant d(\mathbb{P}_p(A), \mathbb{P}_{p'}(A))$$

where $d(x, y) = x \log(x/y) + (1-x)\log((1-x)/(1-y))$.

3.9.4.1 Proof of Theorem 3.4

Consider any arm a. By Assumption 3.3, there exists alternative model p' such that:

$$\mathrm{KL}(p_a, p_{\kappa_{g(a)}+1}) < \mathrm{KL}(p_a, p'_a) < \mathrm{KL}(p_a, p_{\kappa_{g(a)}+1}) + \alpha \text{ and } p'_a < p_{\kappa_{g(a)}+1}$$

Note that in the model p', arm a no longer belongs to the cluster $g(a)$. Let $\hat{M}_{g(a)}$ be the set of arms returned by an algorithm in the $g(a)^{\text{th}}$ cluster. If we define the event $A = \{a \in \hat{M}_{g(a)}\} \in \mathcal{F}_\tau$, then by definition, for any δ-PAC algorithm, $\mathbb{P}_p(A) \geqslant 1-\delta$ and $\mathbb{P}_{p'}(A) \leqslant \delta$. Letting $N_a(\tau)$ denote the number of pulls of arm a by time τ, we have by Lemma 3.12 and the monotonicity of $d(x, y)$ that

$$\mathrm{KL}(p_a, p'_a)\mathbb{E}_p[N_a(\tau)] \geqslant d(1-\delta, \delta) \geqslant \log(\tfrac{1}{2.4\delta})$$

where we use the property that for $x \in [0,1]$, $d(x, 1-x) \geqslant \log \frac{1}{2.4x}$. This gives us that

$$\mathbb{E}_p[N_a(\tau)] \geqslant \frac{1}{\mathrm{KL}(p_a, p_{\kappa_{g(a)}+1}) + \alpha} \log\left(\frac{1}{2.4\delta}\right)$$

Letting $\alpha \to 0$, we get

$$\mathbb{E}_p[N_a(\tau)] \geqslant \frac{1}{\mathrm{KL}(p_a, p_{\kappa_{g(a)}+1})} \log\left(\frac{1}{2.4\delta}\right) \tag{3.14}$$

Similarly, by considering an alternative model p'' such that

$$\mathrm{KL}(p_a, p_{\kappa_{g(a)-1}}) < \mathrm{KL}(p_a, p''_a) < \mathrm{KL}(p_a, p_{\kappa_{g(a)-1}}) + \alpha \text{ and } p''_a > p_{\kappa_{g(a)-1}}$$

we get

$$\mathbb{E}_p[N_a(\tau)] \geqslant \frac{1}{\mathrm{KL}(p_a, p_{\kappa_{g(a)-1}})} \log\left(\frac{1}{2.4\delta}\right) \tag{3.15}$$

From Eq. (3.14), Eq. (3.15), and the definition of $\Delta_\kappa^{\mathrm{KL}}(a)$ in Eq. (3.11), we get that

$$\mathbb{E}_p[N_a(\tau)] \geqslant \frac{1}{\Delta_\kappa^{\mathrm{KL}}(a)} \log\left(\frac{1}{2.4\delta}\right)$$

Summing over all the arms yields the required bound for $\mathbb{E}_p[\tau] = \sum_{a=1}^{K} \mathbb{E}_p[N_a(\tau)]$.

3.10 Extension: Maximum-gap Identification

3.10.1 Introduction

In the coarse ranking setting considered so far in this chapter, the experimenter had to input the cluster boundaries. This simplest instance of this problem is the top-k best arms identification problem, where the experimenter specifies k. If the means of the k^{th} and $(k+1)^{\text{st}}$ arms are very close to each other but there exists a large gap between the means of the $(k+1)^{\text{st}}$ and $(k+2)^{\text{nd}}$ arms, then a better place to separate the clusters in many applications is between arms $(k+1)$ and $(k+2)$.

This motivates the following question: what is the *best* separation of arms into two clusters? If the means are known, a natural way to cluster them is to sort them, find the location of the maximum gap between two adjacent means, and define the two clusters to be the distributions to the left and right of this maximum gap. In the bandit setting the means are unknown and can only be estimated by sampling from the arm's reward distributions. Motivated by this, the problem we study in this section is that of identifiyng the largest gap between the arm means by sampling from their reward distributions.

3.10.2 Setting

Consider a multi-armed bandit with K arms. Each arm a corresponds to a Bernoulli distribution with unknown mean p_a, denoted by $\mathcal{B}(p_a)$. A draw/pull from arm a yields a reward from distribution $\mathcal{B}(p_a)$. Without loss of generality, assume the arms are numbered so that $p_1 > p_2 \cdots > p_K$.

For $i \in [K-1]$, we define the gap between consecutive arms i and $i+1$ to be the difference between their means $(p_i - p_{i+1})$. A natural clustering of the arms into 2 clusters is $C_1 = \{1, \dots, m\}$ and $C_2 = \{m+1, \dots, K\}$,

$$m = \underset{i \in [K-1]}{\arg\max}\,(p_i - p_{i+1}) \tag{3.16}$$

is the location of the largest gap. Our objective in this paper is to design a strategy which given a probability of error $\delta > 0$, samples the arms and upon stopping partitions $[K]$ into two clusters $\hat{C}_1$ and $\hat{C}_2$ such that

$$\mathbb{P}(\hat{C}_1 \neq C_1) \leqslant \delta.$$

This setting is also known as the fixed-confidence setting [**?**], and the goal is to achieve the clustering using as few samples as possible.

3.10.3 Algorithm

We propose an elimination style algorithm. We maintain confidence intervals for the mean of any item and use these to construct upper bounds on the maximum gap *around each item*. We also maintain a lower bound on the overall maximum gap. We stop sampling an item as soon as the upper bound on its maximum gap is smaller than the lower bound.

Let $T_a(t)$ denote the number of times arm a has been played up to time t, and let $\hat{p}_a(t)$ denote its empirical mean at time t. We compute confidence intervals using a function $\beta_\delta(T_a(t))$, and set

$$\begin{aligned} l_a(t) &= \hat{p}_a(t) - \sqrt{\frac{\beta_\delta(T_a(t))}{T_a(t)}}, \\ r_a(t) &= \hat{p}_a(t) + \sqrt{\frac{\beta_\delta(T_a(t))}{T_a(t)}}. \end{aligned} \tag{3.17}$$

The function $\beta_\delta(T_a(t))$ is chosen so that

$$\mathbb{P}(\forall\, t \in \mathbb{N}, \forall\, a \in [K], p_a \in [l_a(t), r_a(t)]) \geqslant 1 - \delta, \tag{3.18}$$

and we discuss its exact form in Remark 3.13.

Arm a has a left gap and a right gap, and we construct separate upper bounds on each of these gaps. Let $U_a^l(t)$ and $U_a^r(t)$ denote the upper bound on the left and right gap of arm a. These can be computed as

$$\begin{aligned} U_a^l(t) &= r_a(t) - \max_{b \in [K]: r_b(t) \leqslant l_a(t)} l_b(t), \\ U_a^r(t) &= \min_{b \in [K]: l_b(t) \geqslant r_a(t)} r_b(t) - l_a(t). \end{aligned} \tag{3.19}$$

The upper bounds in the above equation are illustrated in Figure Fig. 3.7, where the confidence intervals for p_a are indicated by parentheses. For the left gap of arm a, we focus on arms that are with high confidence to the left of a based on current confidence intervals. These are all arms b whose right bound is strictly less than

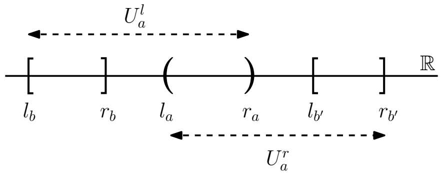


Figure 3.7: Confidence intervals for three arms plotted on the real number line. The argument t is suppressed for brevity. The top dashed line indicates the upper bound for the left gap for arm a which is obtained by the choice of arm b to its left. The bottom dashed line indicates the upper bound for the right gap for arm a which is obtained by the choice of arm b' to its right.

the left bound of a. We can compute trivial upper bounds on the left and right gaps of a by computing the distance to the leftmost and rightmost confidence intervals respectively.

$$\begin{aligned} U_a^L(t) &= r_a(t) - \min_{i \in [K]: i \neq a} l_i(t), \\ U_a^R(t) &= \max_{i \in [K]: i \neq a} r_i(t) - l_a(t) \end{aligned} \tag{3.20}$$

We can combine these as follows to compute an an upper bound $U_a(t)$ on the maximum gap around arm a.

$$U_a(t) = \max\left\{ \min\left\{U_a^l(t), U_a^L(t)\right\} \min\left\{U_a^r(t), U_a^R(t)\right\}\right\} \tag{3.21}$$

To calculate the lower bound on the maximum gap, we sort the items according to their empirical means, and find partitions of items that are clearly separated in terms of their confidence intervals. We explain the computation next. At time t, let (i) denote the item with the i^{th}-largest empirical mean, i.e.,

$$\hat{p}_{(1)}(t) \geqslant \hat{p}_{(2)}(t) \geqslant \cdots \geqslant \hat{p}_{(K)}(t).$$

For $k \in [K-1]$, we have a separation at arm (k) if

$$\min_{a \in \{(1), \dots, (k)\}} l_a(t) > \max_{a \in \{(k+1), \dots, (K)\}} r_a(t),$$

Algorithm 3 `MaxGapElim`

```
Active set $A = [K]$
for $t = 1, 2, \ldots$ do
    for $a \in A$ do
        Sample arm $a$ for $a \in A$.
        Compute bound $[l_a(t), r_a(t)]$ for $p_a$ using (3.17).
    end for

    for $a \in A$ do
        Compute upper bound $U_a(t)$ on max gap around $a$ using (3.21).
    end for
    Compute lower bound $L(t)$ on maximum gap using (3.22).

    // Elimination
    for $a \in A$ do
        if $U_a(t) \leqslant L(t)$ then
            $A = A \setminus a$
        end if
    end for
    if $|A| \leqslant 2$ then
        Return clusters according to the empirical means of the arms and the
        maximum gap.
    end if
end for
```

and each such separation gives us a lower bound on the maximum gap. The best lower bound is then computed as

$$L(t) = \max_{k \in [K-1]} \left(\min_{a \in \{(1), \ldots, (k)\}} l_a(t) - \max_{a \in \{(k+1), \ldots, (K)\}} r_a(t) \right)_+ , \tag{3.22}$$

which is zero if the arms cannot be separated into two clusters with with disjoint confidence intervals.

Remark 3.13. *A simple choice for* β_δ *is* $\beta_\delta(s) = 2\log(cKs^2/\delta)$ *[Even-Dar et al., 2006] for which* (3.18) *holds by Hoeffding's inequality and union bounds, and this is the function* *we use in our analysis. We refer the reader to Garivier [2013], Jamieson et al. [2014] for tighter confidence intervals.*

3.10.4 Analysis

Recall that (without loss of generality) the means are ordered as $p_1 > p_2 \cdots > p_K$. Let the maximum gap exist between arms m and $(m+1)$, and let $\Delta_{\max} = p_m - p_{m+1}$.

We use the notation $(x)_+ = \max\{x, 0\}$. Similarly, if $p_i > p_j$, we denote the gap between them by $\Delta_{j,i} = p_i - p_j$.

3.10.4.1 Accuracy

Theorem 3.14. `MaxGapElim` *returns the correct clustering with probability* $1 - \delta$.

Proof. `MaxGapElim` returns the wrong clustering if the arms with the maximum gap are eliminated from the active set A. We show that this cannot happen if the good event (3.18) holds. The probability that (3.18) does not hold is bounded by δ by Remark 3.13.

Assume (3.18) holds. Let the true maximum gap exist between arms m and $m+1$. Without loss of generality assume to the contrary that arm m is eliminated from A. This happens if $U_m(t) < L(t)$ at some time t. We show that this leads to a contradiction.

We claim that if (3.18) holds, then $p_m - p_{m+1} \leqslant U_m(t)$.

Recall that $L(t)$ is computed using (3.22), and let (s) be the location of the separator in (3.22). Let a be such that $a \in \{(1), \dots, (s)\}$ and $a+1 \in \{(s+1), \dots, (K)\}$. We have that

$$\begin{aligned} p_m - p_{m+1} &\overset{(a)}{\leqslant} U_m(t) \\ &\overset{(b)}{\leqslant} L(t) \\ &\overset{(c)}{\leqslant} l_a(t) - r_{a+1}(t) \\ &\leqslant p_a - p_{a+1}, \end{aligned}$$

where (c) holds because $L(t)$ is the minimum gap between a left confidence interval in $\{(1), \ldots, (s)\}$ and a right confidence interval in $\{(s+1), \ldots, (K)\}$.

This contradicts the fact that $p_m - p_{m+1}$ is the largest gap. $\square$

3.10.4.2 Sample Complexity

For $a \neq 1$, define

$$\Delta_a^r = \max\Big\{ \max_{j:p_j > p_a} \left(\min\{\Delta_{a,j}/4, \left((\Delta_{\max} - \Delta_{a,j})/8\right)\}\right), \\ \left((\Delta_{\max} - \Delta_{a,1})/8\right)\Big\} \tag{3.23}$$

and for $a \neq K$, define

$$\Delta_a^l = \max\Big\{ \max_{j:p_j < p_a} \left(\min\{\Delta_{a,j}/4, \left((\Delta_{\max} - \Delta_{j,a})/8\right)\}\right), \\ \left((\Delta_{\max} - \Delta_{a,K})/8\right)\Big\}. \tag{3.24}$$

We use these to define the gaps which characterize the sample complexity of `MaxGapElim` . Define

$$\Delta_a = \begin{cases} \Delta_a^l & a = 1 \\ \Delta_a^r & a = K \\ \min\{\Delta_a^l, \Delta_a^r\} & \text{otherwise} \end{cases} \tag{3.25}$$

where Δ_a^l, Δ_a^r are defined in (3.24), (3.23).

Theorem 3.15. *With probability at least $1 - \delta$, the sample complexity of `MaxGapElim` is bounded by*

$$H = \sum_{\substack{a \in [K]: \\ a \notin \{m, m+1\}}} \frac{\log(K/\delta\Delta_a)}{\Delta_a^2}$$

where Δ_a is defined in (3.25).

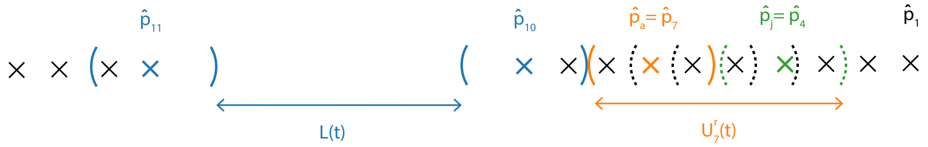


Figure 3.8: Arm a is eliminated when an helper arm j is found

Proof. Arm a is eliminated in `MaxGapElim` when $U_a(t) < L(t)$, where $U(t)$ is defined in (3.21) as the maximum of two terms. Lemma 3.17 and Lemma 3.18 characterize the sufficient condition on c_t for each term to be less than $L(t)$. Since we want both terms to be less than $L(t)$, the sufficient condition we obtain for arm a to be eliminated is $c \leqslant \Delta_a$. The required result then follows from Theorem 3 in Even-Dar et al. [2006], which states that $c_t \leqslant x$ holds when $t = O\left(\frac{\log(K/\delta x)}{x^2}\right)$. □

3.10.4.3 Discussion

We briefly outline in this section the derivation of the gap term in (3.23).

Fig. 3.8 denotes the confidence intervals at some point during the run of `MaxGapElim` . A lower bound on the gap $L(t)$ can be computed between the left and right confidence bounds of arms 10 and 11 respectively as shown. Consider the computation of the upper bound $U_7^r(t)$ on the right gap of arm $a = 7$. The confidence intervals of arms 6 and 5 suggest that these arms can lie to the left of arm 7 and hence these cannot be used to compute the upper bound on the right gap. Arm 4 however lies to the right of arm 7 with high probability, and we can set the upper bound $U_7^r(t) = r_4(t) - l_7(t)$. As soon as $U_7^r(t) < L(t)$, we can remove arm 7 from the active set.

Ignoring the left gap for simplicity, an arm a is removed from the active set as soon as `MaxGapElim` finds an arm j that satisfies two properties: 1) the confidence interval of arm j is disjoint from that of arm a, and 2) the upper bound $U_a^r(t) = r_j(t) - l_a(t) < L(t)$. The first of these conditions gives rise to the term $\Delta_{a,j}/4$ in (3.23), and the second condition gives rise to the term $(\Delta_{\max} - \Delta_{a,j})/8$ in (3.23). Since any arm j that satisfies these conditions can be used to eliminate arm a, we take the maximum over all arms j to yield the smallest sample complexity for arm a.

If the arm j is further to the right, the upper bound $U^r_7(t) = r_4(t) - l_7(t)$ will be larger than $L(t)$ and thus arm 7 cannot be eliminated. Thus the number of times an arm a has to be sampled does not depend only on its own gap. It also depends on whether there is a large gap in the vicinity of arm a. Arm a can have a small gap compared to the maximum gap, but if there is a large gap in its vicinity it will have to be sampled a large number of times. This shows that the sample complexity of the maximum gap identification problem is not the sum of inverse gap of gaps, as one would naively imagine from a reduction to the best arm identification problem.

3.10.4.4 Useful Lemmas

Lemma 3.16. *If the good event* (3.18) *holds, then for all* $a \in [K]$, *for all* $t \in \mathbb{N}$,

$$l_a(t) \geqslant p_a - 2c_{T_a(t)} \textit{ and } r_a(t) \leqslant p_a + 2c_{T_a(t)}$$

where $c_s = \sqrt{\frac{\beta_\delta(s)}{s}}$.

Proof. We have

$$\begin{aligned} \hat{p}_a(t) + c_{T_a(t)} &\overset{(a)}{\geqslant} p_a \\ \Rightarrow l_a(t) = \hat{p}_a(t) - c_{T_a(t)} &\geqslant p_a - 2c_{T_a(t)}. \end{aligned}$$

Similarly,

$$\begin{aligned} \hat{p}_a(t) - c_{T_a(t)} &\overset{(a)}{\leqslant} p_a \\ \Rightarrow r_a(t) = \hat{p}_a(t) + c_{T_a(t)} &\leqslant p_a + 2c_{T_a(t)}. \end{aligned}$$

In both the equations above, (a) holds by (3.18). □

Lemma 3.17. *Assume* (3.18) *holds, and consider* $a \notin \{1, m+1\}$. *In* `MaxGapElim` *if* t *is such that* $c_t \leqslant \Delta^r_a$, *then*

$$\min\left\{U^r_a(t), U^R_a(t)\right\} < L(t).$$

Proof. Assume (3.18) holds. We have $c_t < \Delta_a^r < \Delta_{\max}/4$. This implies that

$$\begin{aligned} l_m(t) &\overset{(a)}{\geqslant} p_m - 2c_t = p_{m+1} + \Delta_{\max} - 2c_t \\ &\overset{(a)}{\geqslant} r_{m+1}(t) + \Delta_{\max} - 4c_t \geqslant r_{m+1}(t). \end{aligned} \tag{3.26}$$

where (a) holds by Lemma 3.16.

From (3.26) we have that

$$L(t) \geqslant l_m(t) - r_{m+1}(t) \geqslant \Delta_{\max} - 4c_t \tag{3.27}$$

We show in part I below that if

$$c_t < \gamma_1 = \max_{j:p_j > p_a} \left(\min\{\Delta_{a,j}/4, ((\Delta_{\max} - \Delta_{a,j})/8)\} \right), \tag{3.28}$$

then $U_a^r(t) < L(t)$. We show in part II that if

$$c_t < \gamma_2 = ((\Delta_{\max} - \Delta_{a,1})/8), \tag{3.29}$$

then $U_a^R(t) < L(t)$. From this we conclude that if $c_t < \max(\gamma_1, \gamma_2) = \Delta_a^r$, then $\min\left\{U_a^r(t), U_a^R(t)\right\} < L(t)$, proving the statement of the lemma.

Part 1: $\mathbf{U_a^r(t) < L(t)}$

We assume (3.28) holds. Let arm e be the maximizer in (3.28), i.e.,

$$e = \underset{j:p_j > p_a}{\arg\max} \left(\min\{\Delta_{a,j}/4, ((\Delta_{\max} - \Delta_{a,j})/8)\} \right). \tag{3.30}$$

For any arm j such that $\Delta_{\max} < \Delta_{a,j}$, the inner minimum in (3.30) will be negative. On the other hand, since $a \neq m+1$, there must exist an arm j such that $\Delta_{\max} > \Delta_{a,j}$, and for such an arm j the inner minimum will be positive. Since e is the arm that maximizes the inner minimum, the inner minimum must be positive for e. Thus we have that $\Delta_{\max} > \Delta_{a,e}$. Furthermore, from (3.28) we have that

$$c_t < (\Delta_{\max} - \Delta_{a,e})/8. \tag{3.31}$$

Since $c_t < \Delta_{a,e}$, by following an argument similar to (3.26) we have that $l_e(t) \geqslant r_a(t)$. Hence we have

$$\begin{aligned} U_a^r(t) &\overset{(a)}{\leqslant} r_e(t) - l_a(t) \\ &\overset{(b)}{\leqslant} \Delta_{a,e} + 4c_t \\ &\overset{(c)}{\leqslant} \Delta_{\max} - 4c_t \\ &\overset{(d)}{\leqslant} L(t) \end{aligned}$$

where (a) follows from (3.19), (b) holds from Lemma 3.16, (c) follows by (3.31), and (d) holds by (3.27).

Part 2: $\mathbf{U_a^R(t) < L(t)}$

We assume (3.29) holds. Recall from (3.20) that $U_a^R(t) = \max_{i \in [K]: i \neq a} r_i(t) - l_a(t)$, and let arm e be the maximizer. Hence

$$\begin{aligned} U_a^R(t) &= r_e(t) - l_a(t) \\ &\overset{(a)}{\leqslant} p_e + 2c_t - p_a + 2c_t \\ &\leqslant \Delta_{a,1} + 4c_t \\ &\overset{(b)}{\leqslant} \Delta_{\max} - 4c_t \\ &\overset{(c)}{\leqslant} L(t) \end{aligned}$$

where (a) holds by Lemma 3.16, (b) holds by (3.29), and (c) holds by (3.27). □

Lemma 3.18. *Assume* (3.18) *holds. Then in* `MaxGapElim` *for* $a \neq \{m, K\}$

$$\min\left\{U_a^l(t), U_a^L(t)\right\} < L(t)$$

definitely holds when $c \leqslant \Delta_a^L$*, where* Δ_a^l *is as defined in* (3.24).

Proof. The proof is analogous to the proof of Lemma 3.17. □

3.10.4.5 Experiments

We try the elimination algorithm above, along with a UCB algorithm. The UCB algorithm computes the UCBs on the gaps and plays all arms whose UCB matches the largest UCB. We plot the mistake probabililty as a function of the number of queries, where a mistake is said to occur if the clustering obtained by finding the maximum gap among the empirical means is different from the clustering obtained by finding the maximum gap in the true means.

First we consider an experiment where the number of arms $K = 100$, and the arm rewards are $\mathcal{N}(\cdot, 1)$. We set the means so that there are large gaps of sizes $\Delta_{\max} = 1.2$ and $\Delta_1 = 1.1$ respectively, and 97 gaps of size $\Delta_2 = 0.2$. The mistake probabilities of the non-adaptive algorithm, `MaxGapElim` , and the UCB algorithm we obtain are as follows. We see roughly 10x gains in the number of queries required for UCB as compared to non-adaptive sampling.

In the second experiment, we consider arms with reward distributions $\mathcal{N}(\cdot, 1)$ as before, but we now consider means such that the gaps between the means decrease geometrically. The mistake probabilities of the non-adaptive algorithm, `MaxGapElim` , and the UCB algorithm we obtain are as follows.

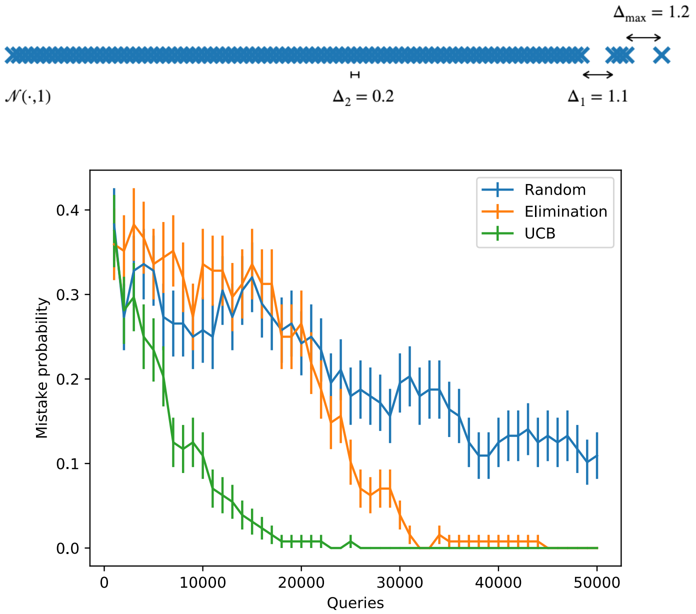


Figure 3.9: (a) Means of arms in Experiment 1. (b) Mistake probabilities of the non-adaptive algorithm, `MaxGapElim` , and UCB algorithms in Experiment 1

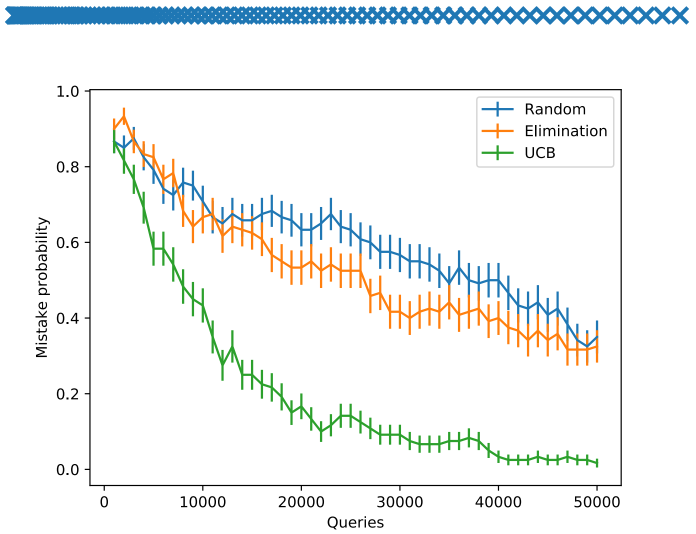


Figure 3.10: (a) Means of arms in Experiment 2. (b) Mistake probabilities of the non-adaptive algorithm, `MaxGapElim` , and UCB algorithms in Experiment 2

4 DCM BANDITS: LEARNING TO RANK WITH MULTIPLE CLICKS

4.1 Introduction

Web pages in search engines are often ranked based on a model of user behavior, which is learned from click data Radlinski and Joachims [2005], Agichtein et al. [2006], Chuklin et al. [2015a]. The cascade model Craswell et al. [2008] is one of the most popular models of user behavior in web search. Kveton et al. [2015a] and Combes et al. [2015a] recently proposed regret-optimal online learning algorithms for the cascade model. The main limitation of the cascade model is that it cannot model multiple clicks. Although the model was extended to multiple clicks Guo et al. [2009b], Chapelle and Zhang [2009], Guo et al. [2009a,b], it is unclear if it is possible to design computationally and sample efficient online learning algorithms for these extensions.

In this work, we propose an online learning variant of the *dependent click model (DCM)* Guo et al. [2009b], which we call *DCM bandits*. The DCM is a generalization of the cascade model where the user may click on multiple items. At time t, our learning agent recommends to the user a list of K items. The user examines the items in the list, from the first item to the last. If the examined item attracts the user, the user clicks on it. This is observed by the learning agent. After the user clicks on the item and investigates it, the user decides whether to leave or examine more items. If the user leaves, the DCM interprets this event as that the user is satisfied, and our learning agent receives a reward of one. If the user scans the list of items until the end and does not leave on purpose, the agent receives a reward of zero. The goal of the learning agent is to maximize its total reward, or equivalently to minimize its cumulative regret with respect to the most satisfactory list of K items. The main challenge of our learning problem is that the agent does not observe whether the user is satisfied. The agent only observes the clicks of the user. This imbalance between the feedback and reward is central to all multi-click generalizations of the cascade model Guo et al. [2009b], Chapelle and Zhang [2009], Guo et al. [2009a,b] and makes learning challenging. This differentiates our setting from that of cascading bandits Kveton et al. [2015a], where the user clicks on at most one item and this click is assumed to be satisfactory.

We make five major contributions. First, we precisely formulate a learning variant of the dependent-click model as a stochastic combinatorial partial monitoring problem. Second, we propose an assumption that the agent knows the order of position-dependent termination probabilities in the DCM. We argue that this assumption is mild. Under this assumption, the optimal list of items can be learned by estimating the attraction probabilities of items from clicks. This is the key idea in our computationally-efficient learning algorithm dcmKL-UCB, which is motivated by KL-UCB. This is our third major contribution. Fourth, we prove gap-dependent upper bounds on the regret of dcmKL-UCB and derive a matching lower bound up to logarithmic factors. The bounds are proved based on a novel reduction to cascading bandits Kveton et al. [2015a] and reflect our intuition that learning from multiple clicks is more sample efficient than learning from a single click. Finally, we comprehensively evaluate our algorithm on both synthetic and real-world problems. Our algorithm outperforms a range of baselines and performs well even when our modeling assumptions are violated.

To simplify exposition, we denote random variables by boldface letters and write $[n]$ instead of $\{1, \dots, n\}$.

4.2 Background

Web pages in search engines are often ranked based on a model of user behavior, which is learned from click data Radlinski and Joachims [2005], Agichtein et al. [2006], Chuklin et al. [2015a]. We assume that the user scans a list of K web pages $A = (a_1, \dots, a_K)$, which we call *items*. The items belong to some *ground set* $E = [L]$, such as the set of all web pages. Many models of user behavior in web search exist Becker et al. [2007], Richardson et al. [2007], Craswell et al. [2008], Chapelle and Zhang [2009], Guo et al. [2009a,b]. We focus on the dependent click model Guo et al. [2009b].

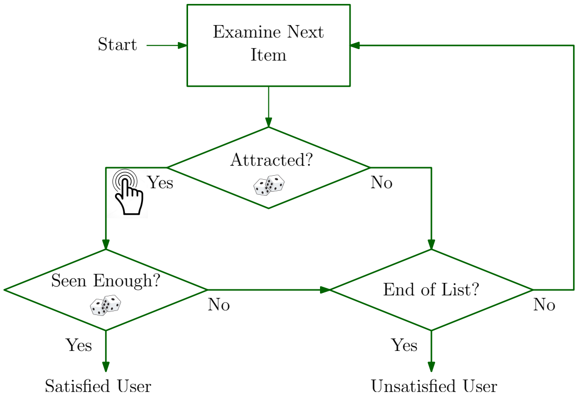


Figure 4.1: Interaction between the user and items in the DCM.

The *dependent click model (DCM)* Guo et al. [2009b] is an extension of the cascade model Craswell et al. [2008] to multiple clicks. The model assumes that the user scans a list of K items $A = (a_1, \dots, a_K) \in \Pi_K(E)$ from the first item a_1 to the last a_K, where $\Pi_K(E) \subset E^K$ is the set of all K-*permutations* of set E. The model is parameterized by *item-dependent attraction probabilities* $\bar{w} \in [0,1]^E$ and *position-dependent termination probabilities* $\bar{v} \in [0,1]^K$. The user interacts in this model as follows. After the user *examines* item a_k, the item *attracts* the user with probability $\bar{w}(a_k)$, *independently* of the other items. If the user is attracted by item a_k, the user clicks on it and *terminates* the search with probability $\bar{v}(k)$. In this case, it is assumed that the user is *satisfied* with item a_k and does not examine the *remaining* items. If the user is not attracted by item a_k, or the user is attracted but does not terminate, the user examines the next item a_{k+1}. This interaction model is visualized in Fig. 4.1. Note the following. The probabilities $\bar{w}(a_k)$ and $\bar{v}(k)$ are *conditional* on that the user examines the item, and that the examined item is attractive, respectively. However, for brevity, we drop "conditional" in the rest of the chapter. Also note that $\bar{v}(k)$ is *not* the probability that the user terminates at position k. The latter depends on the items and positions before position k.

It is easy to see that the probability that the user leaves satisfied given list A is $1 - \prod_{k=1}^{K}(1 - \bar{v}(k)\bar{w}(a_k))$. This objective is maximized by K most attractive items, where the kth most attractive item is placed at the position with the kth highest termination probability.

4.3 DCM Bandits

We propose a learning variant of the dependent click model (Section 4.3.1) and a computationally-efficient algorithm for solving it (Section 4.3.3).

4.3.1 Setting

A *dependent-click model (DCM) bandit* is a tuple $B = (E, P_w, P_v, K)$, where $E = [L]$ is a *ground set* of L items; P_w and P_v are probability distributions over binary hypercubes $\{0,1\}^L$ and $\{0,1\}^K$, respectively; and $K \leqslant L$ is the number of recommended items.

The learning agent interacts with our problem as follows. Let $(\mathbf{w}_t)_{t=1}^n$ be an i.i.d. sequence of n *attraction weights* drawn from P_w, where $\mathbf{w}_t \in \{0,1\}^E$ and $\mathbf{w}_t(e)$ indicates that item e attracts the user at time t. Let $(\mathbf{v}_t)_{t=1}^n$ be an i.i.d. sequence of n *termination weights* drawn from P_v, where $\mathbf{v}_t \in \{0,1\}^K$ and $\mathbf{v}_t(k)$ indicates that the user would terminate at position k if the item at that position was examined and attractive. At time t, the learning agent recommends a list of K items $\mathbf{A}_t = (\mathbf{a}_1^t, \dots, \mathbf{a}_K^t) \in \Pi_K(E)$. The user examines the recommended items in the order in which they are presented, as shown in Fig. 4.1. The learning agent receives a vector of observations $\mathbf{c}_t \in \{0,1\}^K$, which are indicators of the clicks of the user. In particular, $\mathbf{c}_t(k) = 1$ if and only if the user clicks on item $\mathbf{a}_k^t$, the item at position k at time t.

The learning agent receives *reward* $\mathbf{r}_t$, which is *unobserved*. The reward is binary and is one if and only if the user is satisfied with at least one item in $\mathbf{A}_t$. Item e is *satisfactory* at time t when it is attractive, $\mathbf{w}_t(e) = 1$, and its position k leads to termination, $\mathbf{v}_t(k) = 1$. By our assumption, the reward can be expressed as $\mathbf{r}_t = f(\mathbf{A}_t, \mathbf{w}_t, \mathbf{v}_t)$, where we refer to $f : \Pi_K(E) \times [0,1]^E \times [0,1]^K \to [0,1]$ as a *reward* *function* and define it as

$$f(A, w, v) = 1 - \prod_{k=1}^{K} (1 - v(k) w(a_k))\,.$$

for any $A = (a_1, \dots, a_K) \in \Pi_K(E)$, $w \in [0,1]^E$, and $v \in [0,1]^K$. The form of the reward function proves particularly useful in our analysis.

The attraction and termination weights in the DCM are drawn *independently* of each other Guo et al. [2009b]. We adopt the same assumption in our work. In particular, we assume that for any $w \in \{0,1\}^E$ and $v \in \{0,1\}^K$,

$$\begin{aligned} P_w(w) &= \prod\nolimits_{e \in E} \mathrm{Ber}(w(e); \bar{w}(e))\,, \\ P_v(v) &= \prod\nolimits_{k \in [K]} \mathrm{Ber}(v(k); \bar{v}(k))\,, \end{aligned}$$

where $\mathrm{Ber}(\cdot; \theta)$ is a Bernoulli distribution with mean θ. This independence assumption allows us to design a very efficient learning algorithm. Under this assumption, the expected reward for list $A \in \Pi_K(E)$, the probability that at least one item in A is satisfactory, decomposes as

$$\mathbb{E}\left[f(A, \mathbf{w}, \mathbf{v})\right] = 1 - \prod_{k=1}^{K} (1 - \mathbb{E}\left[\mathbf{v}(k)\right] \mathbb{E}\left[\mathbf{w}(a_k)\right]) = f(A, \bar{w}, \bar{v})$$

and depends only on the attraction probabilities of items in A and the termination probabilities $\bar{v}$. An analogous property was useful in the design and analysis of algorithms for cascading bandits Kveton et al. [2015a].

We evaluate the performance of a learning agent by its *expected cumulative regret*

$$R(n) = \mathbb{E}\left[\sum\nolimits_{t=1}^{n} R(\mathbf{A}_t, \mathbf{w}_t, \mathbf{v}_t)\right]\,,$$

where $R(\mathbf{A}_t, \mathbf{w}_t, \mathbf{v}_t) = f(A^*, \mathbf{w}_t, \mathbf{v}_t) - f(\mathbf{A}_t, \mathbf{w}_t, \mathbf{v}_t)$ is the *instantaneous regret* of the agent at time t and

$$A^* = \arg\max_{A \in \Pi_K(E)} f(A, \bar{w}, \bar{v})$$

is the *optimal list* of items, the list that maximized the reward at any time t. It is easy to see that A^* contains K most attractive items, which are ordered such that the kth most attractive item is placed at the position with the kth highest termination probability. For simplicity of exposition, we assume that the optimal solution, as a set, is unique.

4.3.2 Learning Without Accessing Rewards

Learning in DCM bandits seems difficult because the observations $\mathbf{c}_t$ are not sufficient to identify whether the recommended items lead to a reward. Consider the following example. The learning agent recommends items $\mathbf{A}_t = (1, 2, 3, 4)$ and observes $\mathbf{c}_t = (0, 1, 1, 0)$. This feedback can be interpreted in two ways. The first explanation is that item 1 is not attractive, items 2 and 3 are attractive, and that the user does not exit at either positions 2 or 3. The second explanation is that item 1 is not attractive, items 2 and 3 are attractive, and that the user does not exit at position 2, but exits at position 3. In the first case, the learning agent receives no reward; in the second one, it does. Since the reward is not directly observed, DCM bandits can be viewed as an instance of *partial monitoring*. DCM bandits cannot be solved efficiently by existing algorithms for partial monitoring because the action set is combinatorial. Therefore, in this work, we impose an addition mild assumption that allows us to learn efficiently, while avoiding the combinatorial explosion of the action space.

The key idea in our solution is based on the following insight. Without loss of generality, suppose that the termination probabilities are ordered such that $\bar{v}(1) \geqslant \ldots \geqslant \bar{v}(K)$. Then $A^* = \arg\max_{A \in \Pi_K(E)} f(A, \bar{w}, \tilde{v})$ for any vector $\tilde{v} \in [0, 1]^K$ that satisfies $\tilde{v}(1) \geqslant \ldots \geqslant \tilde{v}(K)$. Therefore, the *termination probabilities* do not have to be learned if their *order is known*, which is what we assume from this point on. This assumption is much milder than knowing the probabilities. We show in Section 4.5

that our algorithm performs well even if this order is misspecified.

Before we proceed, we need one more observation. Let

$$\mathbf{C}_t^{\text{last}} = \max\{k \in [K] : \mathbf{c}_t(k) = 1\} \tag{4.1}$$

denote the position of the *last click*, where $\max \emptyset = +\infty$. Then $\mathbf{w}_t(a_k) = \mathbf{c}_t(k)$ for any $k \leqslant \min\left\{\mathbf{C}_t^{\text{last}}, K\right\}$. In other words, $\mathbf{c}_t$ is an observed portion of $\mathbf{w}_t$ up to the last click, and therefore we can use it to learn the attraction probabilities of items in E.

4.3.3 `dcmKL-UCB` Algorithm

We propose a UCB-like algorithm for solving DCM bandits, which we call `dcmKL-UCB`. The algorithm is motivated by `KL-UCB` Garivier and Cappe [2011] and its pseudocode is shown in Algorithm 4. At time t, `dcmKL-UCB` operates in three stages. First, it computes the *upper confidence bounds (UCBs)* $\mathbf{U}_t \in [0,1]^E$ on the attraction probabilities of all items in E. The UCB of item e at time t is

$$\begin{aligned}\mathbf{U}_t(e) = \max\{q \in [w, 1] : w = \hat{\mathbf{w}}_{\mathbf{T}_{t-1}(e)}(e)\,, \\ \mathbf{T}_{t-1}(e) D_{\text{KL}}(w \parallel q) \leqslant \log t + 3 \log\log t\}\,,\end{aligned} \tag{4.2}$$

where $D_{\text{KL}}(p \parallel q)$ is the *Kullback-Leibler (KL) divergence* between Bernoulli random variables with means p and q; $\hat{\mathbf{w}}_s(e)$ is the average of s observed weights of item e; and $\mathbf{T}_t(e)$ is the number of times that item e is observed in t steps. Since $D_{\text{KL}}(p \parallel q)$ increases in q for $q \geqslant p$, our UCB can be computed efficiently. Second, `dcmKL-UCB` recommends a list of K items with largest UCBs:

$$\mathbf{A}_t = \arg\max{}_{A \in \Pi_K(E)} f(A, \mathbf{U}_t, \tilde{v})\,,$$

where $\tilde{v} \in [0,1]^K$ is any vector that satisfies $\tilde{v}(1) > \ldots > \tilde{v}(K)$. The selection of $\mathbf{A}_t$ can be implemented efficiently in $O(L + K \log K)$ time, by placing the item with the k^{th} largest UCB to the k^{th} highest position. After the user provides feedback $\mathbf{c}_t$,

Algorithm 4 UCB-like algorithm for DCM bandits.

```
// Initialization
Observe $\mathbf{w}_0 \sim P_w$
$\forall e \in E : \mathbf{T}_0(e) \leftarrow 1$
$\forall e \in E : \hat{\mathbf{w}}_1(e) \leftarrow \mathbf{w}_0(e)$

for all $t = 1, \dots, n$ do
  for all $e = 1, \dots, L$ do
    Compute UCBs $\mathbf{U}_t(e)$ using (4.2)
  end for

  // Compute recommendation
  Let $\mathbf{A}_t \leftarrow \arg\max_{A \in \Pi_K(E)} f(A, \mathbf{U}_t, \tilde{v})$
  Recommend $\mathbf{A}_t$ and observe clicks $\mathbf{c}_t \in \{0,1\}^K$
  $\mathbf{C}_t^{\text{last}} \leftarrow \max\{k \in [K] : \mathbf{c}_t(k) = 1\}$

  // Update statistics
  $\forall e \in E : \mathbf{T}_t(e) \leftarrow \mathbf{T}_{t-1}(e)$
  for all $k = 1, \dots, \min\left\{\mathbf{C}_t^{\text{last}}, K\right\}$ do
    $e \leftarrow \mathbf{a}_k^t$
    $\mathbf{T}_t(e) \leftarrow \mathbf{T}_t(e) + 1$
    $\hat{\mathbf{w}}_{\mathbf{T}_t(e)}(e) \leftarrow \frac{\mathbf{T}_{t-1}(e)\hat{\mathbf{w}}_{\mathbf{T}_{t-1}(e)}(e) + \mathbf{c}_t(k)}{\mathbf{T}_t(e)}$
  end for
end for
```

dcmKL-UCB updates its estimates of $\bar{w}(e)$ up to position $\min\left\{\mathbf{C}_t^{\text{last}}, K\right\}$, where $\mathbf{C}_t^{\text{last}}$ is the position of the last click (4.1).

We assume that dcmKL-UCB is initialized with one sample of an attraction weight per item. This sample can be generated in most L steps as follows Kveton et al. [2015a]. At time $t \in [L]$, item t is placed at the first position. Because the first position is always examined, $\mathbf{c}_t(1)$ is a random sample of the attraction weight of item t.

4.4 Analysis

In Section 4.4.1, we analyze dcmKL-UCB under the assumptions that all termination probabilities are identical. This simpler case illustrates the key ideas of our proofs. In Section 4.4.2, we consider position-dependent termination probabilities. In Section 4.4.3, we derive a lower bound under the assumption that all termination probabilities are identical. All supplementary lemmas are proved in the Appendix.

For convenience, but without loss of generality, we assume that the items in the ground set E are sorted in decreasing order of their attraction probabilities, $\bar{w}(1) \geqslant \ldots \geqslant \bar{w}(L)$, and that the termination probabilities are sorted in the same way, $\bar{v}(1) \geqslant \ldots \geqslant \bar{v}(K)$. In this case, the *optimal solution* is $A^* = (1, \ldots, K)$ and contains the first K items in E. We say that item e is *optimal* if $e \in [K]$ and that item e is *suboptimal* if $e \in [L] \setminus [K]$. The *gap* between the attraction probabilities of suboptimal item e and optimal item e^*,

$$\Delta_{e,e^*} = \bar{w}(e^*) - \bar{w}(e)\,, \tag{4.3}$$

measures the hardness of discriminating the items. We define the *maximum attraction probability* as $p_{\max} = \bar{w}(1)$ and $\alpha = (1 - p_{\max})^{K-1}$. In practice, we often observe small attraction probabilities, and therefore α is expected to be large, unless K is also large.

The key idea in our analysis is the reduction to cascading bandits Kveton et al. [2015a]. The novelty is in this reduction. As in our model (Section 4.3.1), we define the *cascade reward* over $i \in [K]$ recommended items and the corresponding *expected cascade regret* as

$$f_i(A, w) = 1 - \prod_{k=1}^{i} (1 - w(a_k)) \tag{4.4}$$

$$R_i(n) = \mathbb{E}\left[\textstyle\sum_{t=1}^{n} [f_i(A^*, \mathbf{w}_t) - f_i(\mathbf{A}_t, \mathbf{w}_t)]\right] . \tag{4.5}$$

The cascade regret of dcmKL-UCB can be bounded by adapting the analysis of Kveton et al. [2015a].

Proposition 4.1. *For any $i \in [K]$ and $\varepsilon > 0$, the expected n-step cascade regret of* `dcmKL-UCB` *is bounded as*

$$R_i(n) \leqslant \sum_{e=i+1}^{L} \frac{(1+\varepsilon)\Delta_{e,i}(1+\log(1/\Delta_{e,i}))}{D_{KL}(\bar{w}(e) \,\|\, \bar{w}(i))}(\log n + 3\log\log n) + C\,,$$

where $C = iL\frac{C_2(\varepsilon)}{n^{\beta(\varepsilon)}} + 7i\log\log n$, and $C_2(\varepsilon)$ and $\beta(\varepsilon)$ are defined in Garivier and Cappe [2011].

Proof. The proof is the same as that of Theorem 3 in Kveton et al. [2015a] for the following reason. Our confidence radii have the same form as those in `CascadeKL-UCB`; and for any $\mathbf{A}_t$ and $\mathbf{w}_t$, `dcmKL-UCB` is guaranteed to observe at least as many entries of $\mathbf{w}_t$ as `CascadeKL-UCB`. □

To simplify the presentation of our proofs, we introduce the "*or* function" $\omega : [0,1]^K \to [0,1]$, defined as $\omega(x) = 1 - \prod_{k=1}^{K}(1-x_k)$. For any vectors x and y in the same Euclidean space, we say that $x \geqslant y$ if $x_k \geqslant y_k$ for all indices k. We denote the component-wise product of x and y by $x \odot y$ and the restriction of x to the elements in A by $x|_A$. The latter has a lower precedence than $\odot$. Then, our objective becomes $f(A, \bar{w}, \bar{v}) = \omega(\bar{w} \odot \bar{v}|_A)$.

4.4.1 Upper Bound for Equal Termination Probabilities

Our first upper bound on the regret of `dcmKL-UCB` is under the assumption that all termination probabilities are equal. The next two lemmas relate our objective to a linear function, and they comprise the key steps in our proofs.

Lemma 4.2. *Let $x, y \in [0,1]^K$ satisfy $x \geqslant y$. Then*

$$\omega(x) - \omega(y) \leqslant \textstyle\sum_{k=1}^{K} x_k - \sum_{k=1}^{K} y_k\,.$$

Lemma 4.3. *Let $x, y \in [0, p_{\max}]^K$ satisfy $x \geqslant y$. Then*

$$\alpha\left[\textstyle\sum_{k=1}^{K} x_k - \sum_{k=1}^{K} y_k\right] \leqslant \omega(x) - \omega(y)\,,$$

where $\alpha = (1 - p_{\max})^{K-1}$.

Now we present the main result of this section.

Theorem 4.4. *Let* $\bar{v}(k) = \gamma$ *for all* $k \in [K]$*. For any* $\varepsilon > 0$*, the expected* n*-step regret of* `dcmKL-UCB` *is bounded as*

$$R(n) \leqslant \frac{\gamma}{\alpha} \sum_{e=K+1}^{L} \frac{(1+\varepsilon)\Delta_{e,K}(1+\log(1/\Delta_{e,K}))}{D_{\mathrm{KL}}(\bar{w}(e) \,\|\, \bar{w}(K))}(\log n + 3 \log\log n) + C\,,$$

where $C = \frac{\gamma}{\alpha}\left(KL\frac{C_2(\varepsilon)}{n^{\beta(\varepsilon)}} + 7K\log\log n\right)$*, and* $C_2(\varepsilon)$ *and* $\beta(\varepsilon)$ *are as in Proposition 4.1.*

Proof. Let $\mathbf{R}_t = R(\mathbf{A}_t, \mathbf{w}_t, \mathbf{v}_t)$ be the stochastic regret at time t and

$$\mathcal{H}_t = (\mathbf{A}_1, \mathbf{c}_1, \dots, \mathbf{A}_{t-1}, \mathbf{c}_{t-1}, \mathbf{A}_t) \tag{4.6}$$

be the *history* of the learning agent up to choosing list $\mathbf{A}_t$, the first $t-1$ observations and t actions. By the tower rule, we have $R(n) = \sum_{t=1}^{n} \mathbb{E}\left[\mathbb{E}\left[\mathbf{R}_t \,|\, \mathcal{H}_t\right]\right]$, where

$$\begin{aligned}\mathbb{E}\left[\mathbf{R}_t \,|\, \mathcal{H}_t\right] &= f(A^*, \bar{w}, \bar{v}) - f(\mathbf{A}_t, \bar{w}, \bar{v}) \\ &= \omega(\bar{w} \odot \bar{v}|_{A^*}) - \omega(\bar{w} \odot \bar{v}|_{\mathbf{A}_t})\,.\end{aligned}$$

Now we can apply Lemma 4.2 since $\bar{w} \odot \bar{v}|_{A^*} \geqslant \bar{w} \odot \bar{v}|_{\mathbf{A}_t}$. Moreover, we note that $\bar{v} = \gamma\mathbf{1}$ and then apply Lemma 4.3, and get the following upper bound

$$\begin{aligned}\mathbb{E}\left[\mathbf{R}_t \,|\, \mathcal{H}_t\right] &\leqslant \gamma\left[\sum_{k=1}^{K} \bar{w}(a_k^*) - \sum_{k=1}^{K} \bar{w}(\mathbf{a}_k^t)\right] \\ &\leqslant \frac{\gamma}{\alpha}\left[f_K(A^*, \bar{w}) - f_K(\mathbf{A}_t, \bar{w})\right]\,.\end{aligned}$$

By the definitions of $R(n)$ and $R_K(n)$, and from the above inequality, it follows that

$$R(n) \leqslant \frac{\gamma}{\alpha} \sum_{t=1}^{n} \mathbb{E}\left[f_K(A^*, \bar{w}) - f_K(\mathbf{A}_t, \bar{w})\right] = \frac{\gamma}{\alpha} R_K(n)\,.$$

Finally, we bound $R_K(n)$ using Proposition 4.1. □

4.4.2 General Upper Bound

Our second upper bound on the regret of `dcmKL-UCB` does not make any assumptions on the termination probabilities. However, note that we still assume that `dcmKL-UCB` knows their order. Without loss of generality, we assume that $\bar{v}(1) \geqslant \ldots \geqslant \bar{v}(K)$. To prove our next upper bound, we need a generalization of Lemma 4.2.

Lemma 4.5. *Let $x \in [0,1]^K$ and x' be the permutation of x whose elements are in a decreasing order, $x'_1 \geqslant \ldots \geqslant x'_K$. Let $c \in [0,1]^K$ be another vector whose elements are in a decreasing order. Then*

$$\omega(c \odot x') - \omega(c \odot x) \leqslant \textstyle\sum_{k=1}^K c_k x'_k - \sum_{k=1}^K c_k x_k \,.$$

Now we present our most general upper bound.

Theorem 4.6. *Let $\bar{v}(1) \geqslant \ldots \geqslant \bar{v}(K)$. For any $\varepsilon > 0$, the expected n-step regret of `dcmKL-UCB` is bounded as*

$$R(n) \leqslant (1+\varepsilon) \sum_{i=1}^{K} \frac{\bar{v}(i) - \bar{v}(i+1)}{\alpha} \times$$
$$\sum_{e=i+1}^{L} \frac{\Delta_{e,i}(1 + \log(1/\Delta_{e,i}))}{D_{\mathrm{KL}}(\bar{w}(e) \,\|\, \bar{w}(i))} (\log n + 3 \log\log n) + C\,,$$

where $C = \left(\sum_{i=1}^K \frac{i[\bar{v}(i) - \bar{v}(i+1)]}{\alpha}\right)\left(L\frac{C_2(\varepsilon)}{n^{\beta(\varepsilon)}} + 7\log\log n\right)$, $\bar{v}(K+1) = 0$, and $C_2(\varepsilon)$ and $\beta(\varepsilon)$ are as in Proposition 4.1.

Proof. Let $\mathbf{R}_t$ and $\mathcal{H}_t$ be defined as in the proof of Theorem 4.4. The main challenge in this proof is that we cannot apply Lemma 4.2 to bound $\mathbb{E}[\mathbf{R}_t \mid \mathcal{H}_t]$ because it may happen that $\bar{v}(k)\bar{w}(\mathbf{a}_k^t) > \bar{v}(k)\bar{w}(a_k^*)$ for some $k \in [K]$. To overcome this problem, we rewrite $\mathbb{E}\left[\mathbf{R}_t \,|\, \mathcal{H}_t\right]$ as

$$\begin{aligned}\mathbb{E}\left[\mathbf{R}_t \,|\, \mathcal{H}_t\right] = {} & [\omega(\bar{w} \odot \bar{v}|_{A^*}) - \omega(\bar{w} \odot \bar{v}|_{\mathbf{A}_t'})] + \\ & [\omega(\bar{w} \odot \bar{v}|_{\mathbf{A}_t'}) - \omega(\bar{w} \odot \bar{v}|_{\mathbf{A}_t})]\,,\end{aligned}$$

where $\mathbf{A}_t'$ is the permutation of $\mathbf{A}_t$ where all items are in the order of decreasing attraction probabilities. We bound the first term using Lemma 4.2 and then the second term using Lemma 4.5 to get that

$$\begin{aligned}\mathbb{E}\left[\mathbf{R}_t \,|\, \mathcal{H}_t\right] & \leqslant \sum_{k=1}^{K} \bar{v}(k)(\bar{w}(a_k^*) - \bar{w}(\mathbf{a}_k^t)) \\ & = \sum_{i=1}^{K} [\bar{v}(i) - \bar{v}(i+1)] \sum_{k=1}^{i} (\bar{w}(a_k^*) - \bar{w}(\mathbf{a}_k^t))\,,\end{aligned}$$

where we defined $\bar{v}(K+1) = 0$. Now we bound each $\sum_{k=1}^{i} \bar{w}(a_k^*) - \bar{w}(\mathbf{a}_k^t)$ by Lemma 4.3 to get, via the definitions of $R(n)$ and $R_i(n)$ that

$$R(n) \leqslant \sum_{i=1}^{K} \frac{\bar{v}(i) - \bar{v}(i+1)}{\alpha} R_i(n)\,.$$

Finally, we bound the terms $R_i(n)$ using Proposition 4.1. □

Note that when $\bar{v}(k) = \gamma$ for all $k \in [K]$, the above upper bound reduces to that in Theorem 4.4.

4.4.3 Lower Bound

Our lower bound is derived on the following problem. The ground set are L items $E = [L]$. A subset of these items $A^* \subseteq \Pi_K(E)$ are optimal. The attraction probabilities of items are defined as

$$\bar{w}(e) = \begin{cases} p\,, & e \in A^*\,; \\ p - \Delta\,, & \text{otherwise}\,, \end{cases} \tag{4.7}$$

where p is a common attraction probability of the optimal items and $\Delta \in (0, p)$ is the gap between the attraction probabilities of optimal and suboptimal items. We assume that all termination probabilities are identical, $\bar{v} = \gamma\mathbf{1}$. We denote our problem by $B_{\text{LB}}(L, A^*, p, \Delta, \gamma)$; and parameterize it by L, A^*, p, Δ, and γ. The key step in our analysis is the following lemma.

Lemma 4.7. *Let* $x, y \in [0,1]^K$ *satisfy* $x \geqslant y$. *Let* $\gamma \in [0,1]$. *Then* $\omega(\gamma x) - \omega(\gamma y) \geqslant \gamma[\omega(x) - \omega(y)]$.

Our lower bound is proved for a class of consistent algorithms Lai and Robbins [1985]. The algorithm is *consistent* if for any DCM bandit, any suboptimal item e, and any $\alpha > 0$, $\mathbb{E}\left[\mathbf{T}_n(e)\right] = o(n^\alpha)$, where $\mathbf{T}_n(e)$ is the number of times that item e is recommended in n steps. We also assume that the algorithms observe all entries of $\mathbf{w}_t$ in $\mathbf{A}_t$, $\mathbf{w}_t(\mathbf{a}_k^t)$ for any $k \in [K]$. This is at least as much feedback as in `dcmKL-UCB`.

Theorem 4.8. *For any DCM bandit* B_{LB} *(defined above), the regret of any consistent algorithm that observes all entries of* $\mathbf{w}_t$ *in* $\mathbf{A}_t$ *is bounded from below as*

$$\liminf_{n \to \infty} \frac{R(n)}{\log n} \geqslant \gamma\alpha \frac{(L-K)\Delta}{D_{\text{KL}}(p - \Delta \,\|\, p)}\,.$$

Proof. The key idea of the proof is to reduce our problem to semi-bandits through cascading bandits. First, note that by the tower rule and Lemma 4.7, the n-step regret in DCM bandits is bounded from below as

$$R(n) \geqslant \gamma\mathbb{E}\left[\sum_{t=1}^{n} \left(f_K(A^*, \mathbf{w}_t) - f_K(\mathbf{A}_t, \mathbf{w}_t)\right)\right]\,.$$

Moreover, by the tower rule and Lemma 4.3, we can bound the n-step regret in cascading bandits from below as

$$\begin{aligned} R(n) &\geqslant \gamma\alpha\mathbb{E}\left[\sum_{t=1}^{n}\left(\sum_{k=1}^{K}\mathbf{w}_t(a_k^*) - \sum_{k=1}^{K}\mathbf{w}_t(\mathbf{a}_k^t)\right)\right] \\ &= \gamma\alpha\Delta\sum_{e=K+1}^{L}\mathbb{E}\left[\mathbf{T}_n(e)\right] , \end{aligned}$$

where the last step is based on the fact that all gaps are Δ. By the same argument as in Lai and Robbins [1985], the consistency of the algorithm implies that

$$\liminf_{n\to\infty}\frac{\mathbb{E}\left[\mathbf{T}_n(e)\right]}{\log n} \geqslant \frac{\Delta}{D_{\mathrm{KL}}(p-\Delta \parallel p)}$$

for any suboptimal item e. Now we chain the above two inequalities and get our claim. □

4.4.4 Discussion

We prove gap-dependent upper bounds on the n-step regret of `dcmKL-UCB` under the assumptions that the termination probabilities are identical (Theorem 4.4) and that their order is known (Theorem 4.6). Both bounds are $O(\log n)$, linear in the number of items L, and they improve as the number of recommended items K increases. The bound in Theorem 4.4 is linear in γ, a common termination probability of all positions. Smaller γ results in more clicks. Therefore, we essentially show that the regret decreases with more information, clicks, which is in line with our intuition.

Let us now discuss the tightness of our upper bounds. In particular, consider the problem $B_{\mathrm{LB}}(L, A^* = [K], p = 1/K, \Delta, \gamma)$ of Section 4.4.3. In this setting, $\alpha \geqslant 1/e$ and $1/\alpha \leqslant e$. Therefore, the asymptotic lower bound in Theorem 4.8 and the upper

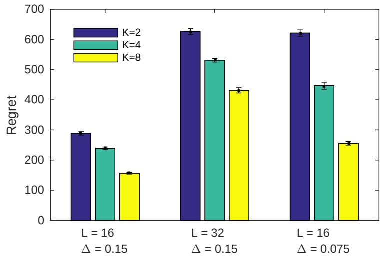


Figure 4.2: The n-step regret of dcmKL-UCB on the problem in Section 4.5.1 in $n = 10^5$ steps. All results are averaged over 20 runs.

bound in Theorem 4.4 respectively reduce to

$$\Omega\left(\gamma(L-K)\frac{\Delta}{D_{KL}(p-\Delta \parallel p)}\log n\right) \tag{4.8}$$

$$O\left(\gamma(L-K)\frac{\Delta(1+\log(1/\Delta))}{D_{KL}(p-\Delta \parallel p)}\log n\right) . \tag{4.9}$$

Note that the bounds match up to $\log(1/\Delta)$.

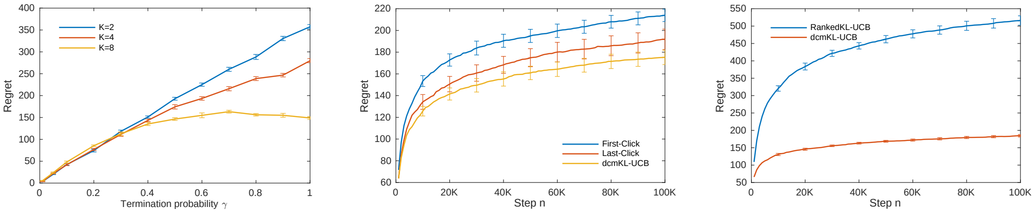
4.5 Experiments

We conduct three experiments. In Section 4.5.1, we validate the upper bound in Theorem 4.4 by showing that the regret of our algorithm scales as suggested there. In Section 4.5.2, we compare our algorithm to several baselines. In Section 4.5.3, we evaluate our algorithm on a real-world dataset.

4.5.1 Regret Bounds

In the first experiment, we validate the behavior of our upper bound in Theorem 4.4. We experiment with the class of problems $B_{LB}(L, [K], p, \Delta, \gamma)$ introduced in Section 4.4.3. We (arbitrarily) choose $p = 0.2$ and $\gamma = 0.8$; and vary L, K, and Δ. We ran our algorithm for $n = 10^5$ steps.

Fig. 4.2 shows the results. We observe two major trends. First, the regret increases when the number of items L increases. Second, the regret decreases when

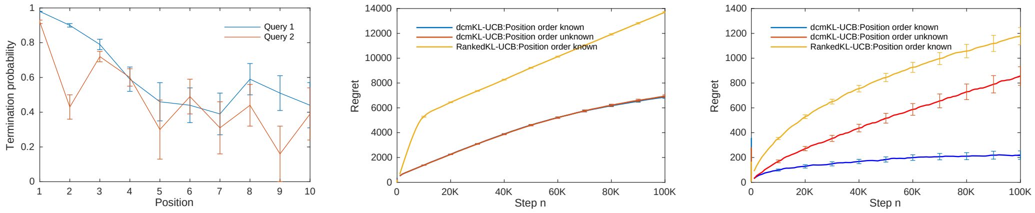


(a)

(b)

(c)

Figure 4.3: **a**. The n-step regret of `dcmKL-UCB` as a function of the termination probability γ. **b**. The n-step regret of the first and last click heuristics, and `dcmKL-UCB` on the problem in Section 4.5.2. **c**. The n-step regret of `RankedKL-UCB` and `dcmKL-UCB` on the same problem.



(a)

(b)

(c)

Figure 4.4: **a**. The termination probabilities in Queries 1 and 2. **b**. The n-step regret in Query 1. **c**. The n-step regret in Query 2.

the number of recommended items K increases. These trends are consistent with the fact that the upper bound in Theorem 4.4 is $O(L-K)$, and the lower bound in Section 4.4.3 is asymptotically $\Omega(L-K)$.

Fig. 4.3a shows the n-step regret of `dcmKL-UCB` as a function of γ. We set $L = 16, p = 0.2, \Delta = 0.15$. We observe that the regret increases linearly with γ when $p < 1/K$, exactly as suggested by the upper bound in Theorem 4.4. This is not surprising. In particular, we note that the key steps in the proof of Theorem 4.4 are based on the linear upper and lower bounds in Lemmas 4.2 and 4.3, and these become tighter as $p \to 0$. Our current analysis is not tight when $p > 1/K$ and we leave this for future work.

4.5.2 First Click, Last Click, and Ranked Bandits

In the second experiment, we compare dcmKL-UCB to two single-click heuristics and a popular algorithm for learning to rank. Both heuristics are based on CascadeKL-UCB Kveton et al. [2015a], which assumes a single clicks. Therefore, in the first heuristic, we modify the feedback $\mathbf{c}_t$ such that it contains only the first click. This heuristic is a conservative extension of CascadeKL-UCB to multiple clicks and we call it First-Click. In the second heuristic, we modify the feedback $\mathbf{c}_t$ such that it contains only the last click. This heuristic can be viewed as learning the satisfaction probabilities of items Kveton et al. [2015a] and we call it Last-Click. Finally, we also compare dcmKL-UCB to a ranked bandit [Radlinski et al., 2008, Slivkins et al., 2013], which we adapt to multiple clicks. The base bandit algorithm of our ranked bandit is KL-UCB, giving rise to the algorithm that we call RankedKL-UCB. The choice of the base algorithm is motivated because all other algorithms are also based on KL-UCB, hence the difference in their regrets will be due to how they use KL-UCB. We experiment with the problem $B_{\text{LB}}(L = 16, A^* = [4], p = 0.2, \Delta = 0.15, \gamma = 0.5)$ from Section 4.4.3.

Fig. 4.3b shows the n-regret of dcmKL-UCB, First-Click, and Last-Click as a function of time n. We observe that the regret of dcmKL-UCB is the lowest among all compared methods. dcmKL-UCB improves upon First-Click and Last-Click, because it does not discard information.

Fig. 4.3c compares RankedKL-UCB and dcmKL-UCB. The regret of RankedKL-UCB is about three times larger than that of dcmKL-UCB. This is not particularly surprising, since the regret in ranked bandits is $O(KL)$ while the regret in dcmKL-UCB is $O(L-K)$ (Section 4.4.4).

4.5.3 Real-World Experiment

In the last experiment, we evaluate dcmKL-UCB on the *Yandex* dataset Yandex, an anonymized search log of 35M million search sessions. Each session contains a query, and the list of displayed documents at positions 1 to 10, and the clicks on those documents. We extract the 5 most frequent queries from our dataset and estimate the parameters of one DCM for each query as in Guo et al. [2009b]. We report results for two queries. In both queries, one document is highly attractive and the attraction probabilities of the remaining documents decay rapidly. The estimated termination probabilities are shown in Fig. 4.4a. The termination probabilities in Query 1 almost decrease with position, while this is not the case in Query 2. We believe that the non-monotonicity is an artifact of the lack of data for training our DCMs.

We study two variants of `dcmKL-UCB`. In one, `dcmKL-UCB` knows the order of the termination probabilities. In the other, it does not and we assume the termination probabilities decrease with position. We also report the regret of `RankedKL-UCB`, which knows the order of the termination probabilities. Fig. 4.4b and Fig. 4.4c show that `dcmKL-UCB` outperforms `RankedKL-UCB`. Furthermore, when the order of the termination probabilities is unknown, the regret of `dcmKL-UCB` increases, though it is still lower than that of `RankedKL-UCB`, even if `RankedKL-UCB` has the advantage of knowing the order of the termination probabilities.

4.6 Related Work

Our work is closely related to *cascading bandits* Kveton et al. [2015a], Combes et al. [2015a], which are learning variants of the cascade model of user behavior Craswell et al. [2008]. Kveton et al. [2015a] proposed a learning algorithm for these problems, `CascadeKL-UCB`; bounded its regret; and proved a matching lower bound. The main limitation of cascading bandits is that they cannot learn from multiple clicks. DCM bandits can be viewed as a generalization of cascading bandits that permits learning from multiple clicks.

Ranked bandits are a popular approach in learning to rank Radlinski et al. [2008], Slivkins et al. [2013]. The key idea in ranked bandits is to model each position in the recommended list as an independent bandit problem, which is then solved by a *base bandit algorithm*. The solutions in ranked bandits are $(1-1/e)$ approximate and the regret is $\Omega(K)$ Radlinski et al. [2008], where K is the number of recommended items. Ranked bandits are analyzed under the assumption that the first clicked item is satisfactory.

Our problem is a partial monitoring problem where we do not observe which items are attractive. Bartok et al. [2012] studied general partial monitoring problems and proposed learning algorithms for solving them. Agrawal et al. [1989] considered a variant of the problem where the reward is observed. The algorithm of Agrawal et al. [1989] cannot be applied to our problem because the reward is unobserved and the algorithm assumes a finite parameter set. The algorithm of Bartok et al. [2012] scales at least linearly with the number of actions, which in our case is $\binom{L}{K}$. Therefore, the algorithm impractical for large L and moderate K. The same applies to the algorithms in Bartók and Szepesvári [2012], Bartók et al. [2014]. Lin et al. [2014] and Kveton et al. [2015b] studied combinatorial partial monitoring but their feedback models are incompatible with our problem.

Our learning problem is combinatorial, we learn the K most satisfactory items out of L. In this sense, our work is related to stochastic combinatorial bandits, which are often studied with linear rewards and semi-bandit feedback Gai et al. [2012a], Chen et al. [2013a], Kveton et al. [2014a, 2015c], Wen et al. [2015], Combes et al. [2015b]. The key differences in our work is that the reward function is non-linear in the unknown parameters and that the feedback is less than semi-bandit as the learning agent does not observe which items are satisfactory.

4.7 Conclusions

In this chapter, we formulate a learning variant of the dependent click model, a popular model of user behavior in web search that can explain multiple clicks. We propose a computationally and sample efficient algorithm for solving it, `dcmKL-UCB`, and prove gap-dependent upper bounds on its regret. The design and analysis of our algorithm are challenging due to the asymmetry between our reward and feedback model. To get around this issue, we propose a reasonable assumption borrowed from the click-modeling literature. The proof of our upper bound then relies on an elegant reduction of the regret to that of the single-click model. Yet the reduction does not lose the opportunity presented by the richer information. We evaluate our algorithm on several problems and show that it outperforms a range of baselines, even when our assumptions are violated.

We leave open several questions of interest. For instance, our upper bound on the regret is linear in the termination probability γ. Fig. 4.3a however shows that this is not tight $p > 1/K$. Thus there is scope to refine our analysis and close the gap between our upper and lower bounds.

To the best of our knowledge, this is the first regret-optimal online learning algorithm for learning to rank with multiple clicks in a cascade-like model. We expect that our work will lead to further exciting new developments in addressing other, perhaps more complex and complete instances of learning to rank under multi-click feedback.

4.8 Appendix

Lemma 4.2. *Let* $x, y \in [0,1]^K$ *satisfy* $x \geqslant y$. *Then:*

$$\omega(x) - \omega(y) \leqslant \sum_{k=1}^{K} x_k - \sum_{k=1}^{K} y_k \,.$$

Proof. Let $x = (x_1, \dots, x_K)$ and:

$$d(x) = \sum_{k=1}^{K} x_k - \omega(x) = \sum_{k=1}^{K} x_k - \left[1 - \prod_{k=1}^{K}(1 - x_k)\right] \,.$$

We prove our claim by showing that $d(x) \geqslant 0$ and $\frac{\partial}{\partial x_i} d(x) \geqslant 0$, for any $x_1, \dots, x_K \in [0,1]$ and $i \in [K]$. First, we show that $d(x) \geqslant 0$ by induction on K. The claim holds trivially for $K = 1$. For any $K \geqslant 2$:

$$d(x) = \sum_{k=1}^{K-1} x_k - \left[1 - \prod_{k=1}^{K-1}(1 - x_k)\right] + \underbrace{x_K - x_K \prod_{k=1}^{K-1}(1 - x_k)}_{\geqslant 0} \geqslant 0\,,$$

where $\sum_{k=1}^{K-1} x_k - \left[1 - \prod_{k=1}^{K-1}(1 - x_k)\right] \geqslant 0$ due to our induction hypothesis. Second, we show that:

$$\frac{\partial}{\partial x_i} d(x) = 1 - \prod_{k \neq i}(1 - x_k) \geqslant 0\,.$$

This concludes our proof. □

Lemma 4.3. *Let* $x, y \in [0, p_{\max}]^K$ *satisfy* $x \geqslant y$. *Then:*

$$\alpha \left[\sum_{k=1}^{K} x_k - \sum_{k=1}^{K} y_k\right] \leqslant \omega(x) - \omega(y)\,,$$

where $\alpha = (1 - p_{\max})^{K-1}$.

Proof. Let $x = (x_1, \dots, x_K)$ and:

$$d(x) = \omega(x) - \alpha \sum_{k=1}^{K} x_k = 1 - \prod_{k=1}^{K}(1 - x_k) - (1 - p_{\max})^{K-1} \sum_{k=1}^{K} x_k\,.$$

We prove our claim by showing that $d(x) \geqslant 0$ and $\frac{\partial}{\partial x_i} d(x) \geqslant 0$, for any $x_1, \dots, x_K \in [0, 1]$ and $i \in [K]$. First, we show that $d(x) \geqslant 0$ by induction on K. The claim holds trivially for $K = 1$. For any $K \geqslant 2$:

$$d(x) = 1 - \prod_{k=1}^{K-1}(1 - x_k) - (1 - p_{\max})^{K-1} \sum_{k=1}^{K-1} x_k + \underbrace{x_K \prod_{k=1}^{K-1}(1 - x_k) - x_K(1 - p_{\max})^{K-1}}_{\geqslant 0} \geqslant 0\,,$$

where $1 - \prod_{k=1}^{K-1}(1 - x_k) - (1 - p_{\max})^{K-1} \sum_{k=1}^{K-1} x_k \geqslant 0$ due to our induction hypothesis and the remainder is non-negative because $1 - x_k \geqslant 1 - p_{\max}$ for any $k \in [K]$. Second, we show that:

$$\frac{\partial}{\partial x_i} d(x) = \prod_{k \neq i}(1 - x_k) - (1 - p_{\max})^{K-1} \geqslant 0\,.$$

This concludes our proof. □

Lemma 4.5. *Let $x \in [0, 1]^K$ and x' be the permutation of x whose elements are in a decreasing order, $x'_1 \geqslant \ldots \geqslant x'_K$. Let $c \in [0, 1]^K$ be another vector whose elements are in a decreasing order. Then:*

$$\omega(c \odot x') - \omega(c \odot x) \leqslant \sum_{k=1}^{K} c_k x'_k - \sum_{k=1}^{K} c_k x_k\,.$$

Proof. Note that our claim is equivalent to proving:

$$1 - \prod_{k=1}^{K}(1 - c_k x'_k) - \left[1 - \prod_{k=1}^{K}(1 - c_k x_k)\right] \leqslant \sum_{k=1}^{K} c_k x'_k - \sum_{k=1}^{K} c_k x_k\,.$$

If $x = x'$, there is nothing to prove. Otherwise, there must exist indices i and j such that $i < j$ and $x_i < x_j$. Let $\widetilde{x}$ be the same vector as x where entries x_i and x_j are exchanged, $\widetilde{x}_i = x_j$ and $\widetilde{x}_j = x_i$. Since $i < j$, $c_i \geqslant c_j$. Let:

$$X_{\text{-}i,\text{-}j} = \prod_{k \neq i,j} (1 - c_k x_k)\,.$$

Then:

$$\begin{aligned}
1-\prod_{k=1}^{K}(1-c_k x'_k)-\left[1-\prod_{k=1}^{K}(1-c_k x_k)\right] &= X_{\text{-i,-j}}\left((1-c_i x_i)(1-c_j x_j)-(1-c_i\widetilde{x}_i)(1-c_j\widetilde{x}_j)\right)\\
&= X_{\text{-i,-j}}\left((1-c_i x_i)(1-c_j x_j)-(1-c_i x_j)(1-c_j x_i)\right)\\
&= X_{\text{-i,-j}}\left(-c_i x_i-c_j x_j+c_i x_j+c_j x_i\right)\\
&= X_{\text{-i,-j}}(c_i-c_j)(x_j-x_i)\\
&\leqslant (c_i-c_j)(x_j-x_i)\\
&= c_i x_j+c_j x_i-c_i x_i-c_j x_j\\
&= c_i\widetilde{x}_i+c_j\widetilde{x}_j-c_i x_i-c_j x_j\\
&= \sum_{k=1}^{K} c_k\widetilde{x}_k-\sum_{k=1}^{K} c_k x_k\,,
\end{aligned}$$

where the inequality is by our assumption that $(c_i-c_j)(x_j-x_i)\geqslant 0$. If $\widetilde{x}=x'$, we are finished. Otherwise, we repeat the above argument until $x=x'$. □

Lemma 4.7. *Let $x, y\in[0,1]^K$ satisfy $x\geqslant y$. Let $\gamma\in[0,1]$. Then:*

$$\omega(\gamma x)-\omega(\gamma y)\geqslant\gamma[\omega(x)-\omega(y)]\,.$$

Proof. Note that our claim is equivalent to proving:

$$\prod_{k=1}^{K}(1-\gamma y_k)-\prod_{k=1}^{K}(1-\gamma x_k)\geqslant\gamma\left[\prod_{k=1}^{K}(1-y_k)-\prod_{k=1}^{K}(1-x_k)\right].$$

The proof is by induction on K. To simplify exposition, we define the following shorthands:

$$X_i=\prod_{k=1}^{i}(1-x_k)\,,\quad X_i^{\gamma}=\prod_{k=1}^{i}(1-\gamma x_k)\,,\quad Y_i=\prod_{k=1}^{i}(1-y_k)\,,\quad Y_i^{\gamma}=\prod_{k=1}^{i}(1-\gamma y_k)\,.$$

Our claim holds trivially for $K = 1$ because:

$$(1 - \gamma y_1) - (1 - \gamma x_1) = \gamma[(1 - y_1) - (1 - x_1)] \,.$$

To prove that the claim holds for any K, we first rewrite $Y_K^\gamma - X_K^\gamma$ in terms of $Y_{K-1}^\gamma - X_{K-1}^\gamma$:

$$\begin{aligned} Y_K^\gamma - X_K^\gamma &= (1 - \gamma y_K) Y_{K-1}^\gamma - (1 - \gamma x_K) X_{K-1}^\gamma \\ &= Y_{K-1}^\gamma - \gamma y_K Y_{K-1}^\gamma - X_{K-1}^\gamma + \gamma y_K X_{K-1}^\gamma + \gamma (x_K - y_K) X_{K-1}^\gamma \\ &= (1 - \gamma y_K)(Y_{K-1}^\gamma - X_{K-1}^\gamma) + \gamma (x_K - y_K) X_{K-1}^\gamma \,. \end{aligned}$$

By our induction hypothesis, $Y_{K-1}^\gamma - X_{K-1}^\gamma \geqslant \gamma(Y_{K-1} - X_{K-1})$. Moreover, $X_{K-1}^\gamma \geqslant X_{K-1}$ and $1 - \gamma y_K \geqslant 1 - y_K$. We apply these lower bounds to the right-hand side of the above equality and then rearrange it as:

$$\begin{aligned} Y_K^\gamma - X_K^\gamma &\geqslant \gamma (1 - y_K)(Y_{K-1} - X_{K-1}) + \gamma (x_K - y_K) X_{K-1} \\ &= \gamma[(1 - y_K) Y_{K-1} - (1 - y_K + y_K - x_K) X_{K-1}] \\ &= \gamma[Y_K - X_K] \,. \end{aligned}$$

This concludes our proof. □

5 CONSERVATIVE EXPLORATION USING INTERLEAVING

5.1 Introduction

Recommender systems are an integral component of many industries, with applications in content personalization, advertising, and page design [Resnick and Varian, 1997, Adomavicius and Tuzhilin, 2015, Broder, 2008]. Multi-armed bandit algorithms provide adaptive techniques for content recommendation. However, although they are theoretically well-understood, they have not been widely adopted in production systems [Cremonesi et al., 2011, Schnabel et al., 2018]. This is primarily due to concerns that the output of the bandit algorithm can be suboptimal or even disastrous, especially when the algorithm explores suboptimal arms. To address this issue, most industries have a default recommendation engine in production that has been well-optimized and tested for many years, and a promising new policy is often evaluated using A/B testing [Siroker and Koomen, 2013], which allocates a small α fraction of the traffic to the new policy. When the utilities of actions are independent, this is a reasonable solution that allows the new policy to be evaluated conservatively.

Many recommendation problems involve *structured actions*, such as sets of recommended movies. In these problems, the total utility of the action can be decomposed into the utilities of items in it, such as individual movies. Therefore, it is conceivable that the new policy could be evaluated in a controlled and principled fashion by *interleaving* items in the new and default actions, instead of dividing the traffic as in A/B testing. As a concrete example, consider the problem of recommending top-K movies to a new visitor [Deshpande and Karypis, 2004]. A company may have a default policy that recommends a fixed set of K movies that performs well, but intends to test a new algorithm that promises to learn better movies. The A/B testing method would show the recommendations of the new algorithm to a visitor with probability α. In the initial stages, the new algorithm is expected to explore a lot to learn, and may hurt engagement with the visitor who is shown a disastrous set of movies, just to learn that these movies are not good. An arguably better approach, which does not hurt any visitor's engagement as much and gathers the same feedback on average, is to show the default well-tested movies *interleaved* with α fraction of new recommendations. A recent study by Schnabel et al. [2018] concluded that this latter approach is in fact better,

> "These findings indicate that for improving recommendation systems in practice, it is preferable to mix a limited amount of exploration into every impression – as opposed to having a few impressions that do pure exploration."

In this chapter, we formalize the above idea and study the general case where actions are *exchangeable*, which is a mathematical formulation of the notion of interleaving. In particular, we study learning variants of maximizing an unknown linear function on an exchangeable action set subject to a conservative constraint.

In our motivating recommendation example, we require that any recommendation is always above a certain baseline quality. The question that we want to answer is *what is the price of being this conservative*? In this work, we answer this question and make five contributions. First, we introduce the idea of *conservative multi-armed bandits in combinatorial action spaces*, and formulate a conservative constraint that addresses the issues raised in Schnabel et al. [2018]. Existing conservative constraints for multi-armed bandit problems do not address this issue, as discussed in Section 5.6. Second, we propose interleaving as a solution, and show how it naturally leads to the idea of *exchangeable* action spaces. We precisely formulate *conservative interleaving bandits*, a constrained online learning problem in exchangeable action spaces. Third, we present *Interleaving Upper Confidence Bound (*`iUCB`*)*, a computationally and sample-efficient algorithm for solving our problem. The algorithm satisfies our conservative constraint by design. Fourth, we prove gap-dependent upper bounds on its expected n-step regret. The bounds are logarithmic in the number of steps n, linear in the number of items L, and increase with the level of conservatism. Finally, we evaluate `iUCB` on both synthetic and real-world problems. In synthetic experiments, we validate an extra factor in our regret bounds, which is the price for being conservative. In real-world experiments, we formulate and solve two top-K recommendation problems. To the best of our knowledge, this is the first work that studies conservatism in combinatorial bandit problems.

5.2 Setting

We formulate our online learning problem as a stochastic combinatorial semi-bandit [Kveton et al., 2015d, Gai et al., 2012b, Chen et al., 2013b], which we review in Section 5.2.1. In Section 5.2.2, we define our notion of conservativeness. In Section 5.2.3, we suggest interleaving as a solution and formulate it mathematically using the notion of exchangeable action spaces. Finally, in Section 5.2.4, we introduce our online learning problem of *conservative interleaving bandits*. To simplify exposition, we write all random variables in bold. We denote $\{1, \dots, K\}$ by $[K]$.

5.2.1 Stochastic Combinatorial Semi-Bandits

A *stochastic combinatorial semi-bandit* [Kveton et al., 2015d, Gai et al., 2012b, Chen et al., 2013b] is a tuple $(E, \mathcal{B}, P)$, where $E = [L]$ is a finite set of L items; $\mathcal{B} \subseteq \Pi_K(E)$ is a set of feasible actions, which is a subset of all sets of size K from E, $\Pi_K(E)$; and P is a probability distribution over a unit cube $[0, 1]^E$.

The learning agent interacts with this problem as follows. Let $(\mathbf{w}_t)_{t=1}^n$ be a sequence of n i.i.d. weights drawn from P, where $\mathbf{w}_t(e)$ is the weight of item $e \in E$ at time t. At time t, the agent takes action $\mathbf{A}_t \in \mathcal{B}$, which is a set of K items from E. The reward for taking the action is $f(\mathbf{A}_t, \mathbf{w}_t)$, where $f(A, w) = \sum_{e \in A} w(e)$ is the sum of the weights of all items in A. After taking action $\mathbf{A}_t$, the agent observes the weight $\mathbf{w}_t(e)$ of each item $e \in \mathbf{A}_t$.

The expected weights of items are defined as $\bar{w} = \mathbb{E}[\mathbf{w}]$. The learning agent is evaluated by its *expected n-step regret* $R(n) = \sum_{t=1}^n \mathbb{E}\left[f(A_*, \bar{w})\right] - \sum_{t=1}^n \mathbb{E}\left[f(\mathbf{A}_t, \bar{w})\right]$, where $A_* = \arg\max_{A \in \mathcal{B}} f(A, \bar{w})$ is the *best action in hindsight*.

Stochastic combinatorial semi-bandits can be used to model top-K recommendation problems as follows. The ground set E is the set of all items that can be recommended, such as movies. The action $A \in \mathcal{B}$ is any set of K movies that can be recommended jointly to the user. The weight of item e at time t, $\mathbf{w}_t(e)$, is an indicator of the click on item e at time t. This interaction model is known as the *document click model* [Chuklin et al., 2015c].

5.2.2 Conservative Constraint

To avoid disastrous actions, which may contain a large number of bad items, we impose a constraint on the actions of the learning agent. This constraint is stated formally below.

Let K denote the number of items in all actions. Let B_0 be the *default baseline action*. Our constraint requires that at any time t, the action $\mathbf{A}_t$ of the learning agent should be comparable to or better than the baseline action B_0, in the sense that most items in $\mathbf{A}_t$ should be at least as good as those in B_0. Mathematically, we require that there exists a bijection $\rho_{\mathbf{A}_t, B_0} : \mathbf{A}_t \to B_0$ such that

$$\sum_{e \in \mathbf{A}_t} \mathbb{1}(\bar{w}(e) \geqslant \bar{w}(\rho_{\mathbf{A}_t, B_0}(e))) \geqslant (1 - \alpha)K \tag{5.1}$$

holds with a high probability at all times $t \in [n]$, where α is a problem-specific *risk tolerance parameter*. In other words, the items in $\mathbf{A}_t$ and B_0 can be matched such that at most α fraction of the items in $\mathbf{A}_t$ has a lower expected reward than the matched items in B_0. We compare (5.1) to other notions of conservatism in the literature in Section 5.6.

5.2.3 Exchangeable Actions

Given an algorithm that explores and suggests new actions that could potentially be disastrous, a natural way to satisfy (5.1) is to *interleave* most items from the default action B_0 with a few items from the new action. This is possible if the set of feasible actions $\mathcal{B} \subseteq \Pi_K(E)$ is *exchangeable*.

Definition 5.1 (Exchangeable set). *Given a set* E*, a set* $\mathcal{B} \subseteq \Pi_{\mathsf{K}}(\mathsf{E})$ *is exchangeable if for any two actions* $A_1, A_2 \in \mathcal{B}$*, there exists a bijection* $\rho_{A_1,A_2} : A_1 \to A_2$ *such that*

$$\forall\, G \subseteq A_1 : A_1 \setminus G \cup \{\rho_{A_1,A_2}(e) : e \in G\} \in \mathcal{B}\,. \tag{5.2}$$

From now on, we assume that all sets of feasible actions $\mathcal{B}$ are exchangeable. We give examples of two exchangeable sets below.

Our first example are top-K movie recommendations from Section 5.1. In this example, E is the set of movies and the exchangeable set $\mathcal{B}$ are all subsets of size K from E. The bijection ρ_{A_1,A_2} between two actions $A_1, A_2 \in \mathcal{B}$ can be any bijection subject to the constraint that common items in A_1 and A_2 are mapped to each other. Formally, ρ_{A_1,A_2} is any bijection $A_1 \to A_2$ such that $\rho_{A_1,A_2}(e) = e$ for any $e \in A_1 \cap A_2$. The set $\mathcal{B}$ in this example is also known as a *uniform matroid* of rank K.

Our second example are diverse movie recommendations. Let E be the set of movies and $\mathcal{P}_1, \dots, \mathcal{P}_K$ be a partition of E, where each $\mathcal{P}_i$ represents a movie genre. Then we define the exchangeable set as

$$\mathcal{B} = \{A \in \Pi_K(E) : a_1 \in \mathcal{P}_1, \dots, a_K \in \mathcal{P}_K\}, \tag{5.3}$$

where $A = \{a_1, \dots, a_K\}$. Based on the above definition, any action $A \in \mathcal{B}$ contains one movie from each genre, and hence is diverse. The bijection ρ_{A_1,A_2} between two actions $A_1, A_2 \in \mathcal{B}$ maps $e \in A_1 \cap \mathcal{P}_i$ to $e' \in A_2 \cap \mathcal{P}_i$ for all $i \in [K]$. The set $\mathcal{B}$ in this example is known as a *partition matroid* of rank K.

We briefly explain how exchangeability leads to interleaving of items and allows conservative exploration. In both movie recommendation examples, we can set A_1 to be the default baseline action and A_2 to be a newly evaluated action. A natural approach to exploring A_2 without violating the conservative constraint in (5.1) is through interleaving, all items in the new action A_2 are explored in $S = 1/\alpha$ steps by taking S interleaved actions. Each *interleaved action* substitutes αK unique items in A_1 for the matched items in A_2. Any such action is feasible by Definition 5.1.

For simplicity of exposition, we make two assumptions on α. First, $1/K \leqslant$

$\alpha \leqslant 1/2$. This boundary condition says that we do not consider extreme non-conservative cases, where the learning agent can explore more than a half of items in a new action A_2; and extreme conservative cases, where the learning agent cannot explore safely at least one item in A_2. Second, we assume that $\alpha K \in \mathbb{N}$. This means that all items in A_2 can be observed once in exactly $S = 1/\alpha$ interleaved actions. If this latter assumption is violated, we suggest that α is set to the maximum value of $\alpha' < \alpha$ that satisfies the assumption. This setting is clearly more conservative and satisfies both of our assumptions.

5.2.4 Conservative Interleaving Bandits

A *conservative interleaving bandit* is variant of a stochastic combinatorial semi-bandit (Section 5.2.1) for conservative exploration. Formally, it is a tuple $(E, \mathcal{B}, P, B_0, \alpha)$, where E, $\mathcal{B}$, and P are defined as in Section 5.2.1; $\mathcal{B}$ is an exchangeable set (Definition 5.1), $B_0 \in \mathcal{B}$ is a default baseline action, and $\alpha \in [0, 1]$ is the risk tolerance parameter in (5.1). We assume that the learning agent knows E, $\mathcal{B}$, B_0, and α; and that the distribution P is unknown.

5.3 Algorithm

Learning in conservative interleaving bandits is non-trivial. For instance, one cannot simply take optimistic actions of existing non-conservative algorithms for combinatorial semi-bandits [Kveton et al., 2014b, Talebi and Proutiere, 2016] and interleave them with $1 - \alpha$ fraction of items from the default baseline action B_0. The regret of this policy would be linear because its actions never converge to the optimal action A^*, unless all items in B_0 are optimal. If this was the case, we would not have a learning problem to start with.

In this section, we introduce our *Interleaving Upper Confidence Bound* (`iUCB`) algorithm, which achieves sublinear regret by continuously improving the default baseline action B_0 with a high probability. We present two variants of the algorithm, `iUCB1` and `iUCB2`. In `iUCB1`, the agent knows the expected rewards of all items in

B_0, $\{\bar{w}(e) : e \in B_0\}$. These rewards may be known if the baseline policy has been deployed before. In iUCB1, the agent does not know the expected rewards of items in B_0. We refer to the common aspects of both algorithms as iUCB.

The pseudocode of both algorithms is in Algorithm 5. We highlight their differences in comments. Recall that K is the number of items in all actions. iUCB operates in rounds, which are indexed by t, and takes S interleaved actions in each round. We assume that iUCB has access to an oracle OPT that returns the most rewarding action for any weight vector $w \in [0,1]^E$. The input to OPT is w. When $\mathcal{B}$ are bases of a *matroid*, as in our examples in Section 5.2.3, OPT is a greedy algorithm for finding the maximum weight basis of a matroid and runs in $O(L \log L)$ time [Edmonds, 1971].

In each round, iUCB has three stages. In the first stage (lines 9–10), iUCB computes high-probability *upper confidence bounds (UCBs)* $\mathbf{U}_t \in (\mathbb{R}^+)^E$ and *lower confidence bounds (LCBs)* $\mathbf{L}_t \in (\mathbb{R}^+)^E$ on the expected rewards of all items. For any item $e \in E$,

$$\begin{aligned} \mathbf{U}_t(e) &= \hat{\mathbf{w}}_{\mathbf{T}_{t-1}(e)}(e) + c_{n,\mathbf{T}_{t-1}(e)}\,, \\ \mathbf{L}_t(e) &= \max\{\hat{\mathbf{w}}_{\mathbf{T}_{t-1}(e)}(e) - c_{n,\mathbf{T}_{t-1}(e)}, 0\}\,, \end{aligned} \tag{5.4}$$

where $\hat{\mathbf{w}}_s(e)$ is the average of the first s observed weights of item e, $\mathbf{T}_t(e)$ is the number of times that item e is observed in the first t steps, and

$$c_{n,s} = \sqrt{1.5 \log(n)/s} \tag{5.5}$$

is the radius of a confidence interval around $\hat{\mathbf{w}}_s(e)$ such that $\bar{w}(e) \in [\hat{\mathbf{w}}_s(e) - c_{n,s}, \hat{\mathbf{w}}_s(e) + c_{n,s}]$ holds with a high probability. We use UCB1 confidence intervals [Auer et al., 2002b] to simplify analysis, but it is possible to use tighter KL-UCB confidence intervals [Garivier and Cappé, 2011].

In line 12, iUCB chooses *decision set* $\mathbf{D}_t$, which is the optimal action with respect to weights $\mathbf{U}_t$, an optimistic estimate of $\bar{w}$. The same approach was used in *Optimistic Matroid Maximization* (OMM) of Kveton et al. [2014b]. However, unlike OMM, iUCB cannot take $\mathbf{D}_t$ because it may not satisfy our conservative constraint in (5.1). We refer to $\mathbf{D}_t$ as a *set* to distinguish it from the actions of `iUCB`.

In the second stage (lines 13–20), `iUCB` computes *baseline set* $\mathbf{B}_t$, which is the optimal action with respect to weights $\mathbf{v}_t$. We refer to $\mathbf{B}_t$ as a *set* to distinguish it from the actions of `iUCB`. The weights $\mathbf{v}_t$ are set as follows. If $e \in B_0$, we set $\mathbf{v}_t(e) = \bar{w}(e)$ when $\bar{w}(e)$ is known, and $\mathbf{v}_t(e) = \mathbf{U}_t(e)$ when it is not. If $e \in E \setminus B_0$, we set $\mathbf{v}_t(e) = \mathbf{L}_t(e)$. This setting guarantees that if any item $e \in E \setminus B_0$ is chosen to $\mathbf{B}_t$ over any item $e' \in B_0$, its expected reward is higher than that of item e' with a high probability. As a result, the baseline is improved.

In the last stage (lines 22–31), `iUCB` takes $S = 1/\alpha$ combined actions of $\mathbf{D}_t$ and $\mathbf{B}_t$, which are guaranteed to be in $\mathcal{B}$ by Definition 5.1. In particular, let $\textbf{æ}_t : \mathbf{B}_t \to \mathbf{D}_t$ be the bijection in Definition 5.1 and $\{\mathbf{B}_t^s\}_{s=1}^S$ be a partition of $\mathbf{B}_t$ into S sets such that $|\mathbf{B}_t^s| = \alpha K$ for all $s \in [S]$. Then we take actions $\mathbf{A}_t = \mathbf{B}_t \setminus \mathbf{B}_t^s \cup \{\textbf{æ}_t(e) : e \in \mathbf{B}_t^s\}$ for $s \in [S]$ sequentially. Since $\mathbf{A}_t$ contains at least $(1-\alpha)K$ baseline items, all of which improve over B_0 with a high probability, the conservative constraint in (5.1) is satisfied.

After each action, `iUCB` updates its sufficient statistics (lines 29–31), which are used to estimate the UCBs and LCBs in the next round.

5.4 Analysis

This section has three subsections. In Section 5.4.1, we prove that `iUCB1` is conservative and bound its regret. The main challenge in our analysis is that we cannot directly apply a UCB-like argument, because the baseline set $\mathcal{B}_t$ is chosen based on lower confidence bounds. In Section 5.4.2, we prove analogous claims for `iUCB2`. In Section 5.4.3, we discuss our theoretical results.

We adopt the following conventions in our analysis. Without loss of generality, we assume that items in E are ordered such that $\bar{w}(1) \geqslant \cdots \geqslant \bar{w}(L)$. The optimal action is A^*, the decision set at time t is $\mathbf{D}_t$, and the baseline set at time t is $\mathbf{B}_t$. Recall that A^*, $\mathbf{D}_t$, and $\mathbf{B}_t$ are the elements of an exchangeable action set $\mathcal{B}$, which is defined in Definition 5.1. At any time t, let $\textbf{ß}_t : A^* \to \mathbf{D}_t$ and $\textbf{œ}_t : \mathbf{D}_t \to \mathbf{B}_t$ be the bijections in Definition 5.1, which are guaranteed to exist. These bijections significantly simplify our analysis, and allow us to decompose the improvements in $\mathbf{D}_t$ and $\mathbf{B}_t$ to those of individual items in them.

For any items e and e' such that $\bar{w}(e') \geqslant \bar{w}(e)$, we define the *gap* as $\Delta_{e,e'} = \bar{w}(e') - \bar{w}(e)$. We also define a "good" event at time t as

$$\mathcal{E}_t = \{\forall\, e \in E : |\bar{w}(e) - \hat{\mathbf{w}}_{\mathbf{T}_{t-1}(e)}(e)| \leqslant c_{n,\mathbf{T}_{t-1}(e)}\}\,, \tag{5.6}$$

which is the event that $\bar{w}(e)$ is in the high-probability confidence interval around $\hat{\mathbf{w}}_{\mathbf{T}_{t-1}(e)}(e)$ for all items e at the beginning of time t.

5.4.1 `iUCB1`: Known Baseline Mean Rewards

First, we show that `iUCB1` is conservative. The proof of this claim is in Section 5.8.1.

Theorem 5.2. *`iUCB1` satisfies* (5.1) *jointly at all times* $t \in [n]$ *with probability of at least* $1 - 2L/(Kn)$.

Then we prove a gap-dependent upper bound on the regret of `iUCB1`. The bound involves two kinds of gaps. For any suboptimal item e, we define its gap from the closest better optimal item,

$$\Delta_{e,\min} = \min_{e^* \in A^* : \Delta_{e,e^*} > 0} \Delta_{e,e^*}\,. \tag{5.7}$$

In addition, for any optimal item e^*, we define its gap from the closest worse suboptimal item,

$$\Delta^*_{e^*,\min} = \min_{e \in E \setminus A^* : \Delta_{e,e^*} > 0} \Delta_{e,e^*}\,. \tag{5.8}$$

Our regret bound is stated below.

Theorem 5.3 (Regret of `iUCB1`)**.** *The expected n-step regret of `iUCB1` is bounded as*

$$(S-1)\left(\sum_{e \in E \setminus A^*} \frac{24}{\Delta_{e,\min}} + \sum_{e^* \in A^*} \frac{12}{\Delta^*_{e^*,\min}}\right)\log n + \sum_{e \in E \setminus A^*} \frac{12}{\Delta_{e,\min}} \log n + c\,,$$

where $S = 1/\alpha$; $\Delta_{e,\min}$ *and* $\Delta^*_{e^*,\min}$ *are defined in* (5.7) *and* (5.8)*, respectively; and* $c = O(SL\sqrt{\log n})$.

Proof. Let $\bar{\mathcal{E}} = \bigcup_{t=1}^{n/S} \bar{\mathcal{E}}_t$ be the event that at least one event $\mathcal{E}_t$ in (5.6) does not occur; and $\bar{\mathcal{E}}$ be its complement, the event that all events $\mathcal{E}_t$ in (5.6) occur. Let $\mathbf{R}_t$ the stochastic regret at time t.

We decompose the expected n-step regret by conditioning on $\mathcal{E}$ and $\bar{\mathcal{E}}$ as

$$R(n) = \mathbb{E}\left[\mathbb{1}\left(\bar{\mathcal{E}}\right)\sum_{t=1}^{n/S}\mathbf{R}_t\right] + \mathbb{E}\left[\mathbb{1}\left(\mathcal{E}\right)\sum_{t=1}^{n/S}\mathbf{R}_t\right] . \tag{5.9}$$

The regret due to the first term in (5.9) is low. In particular, since $P(\bar{\mathcal{E}}) \leqslant 2LK^{-1}n^{-1}$ (Lemma 5.6 in Appendix) and the maximum n-step regret is Kn, the maximum contribution due to the first term is $2L$.

In the rest of the proof, we analyze the second term in (5.9) under event $\mathcal{E}$. The key observation is that the expected regret at time t decomposes as

$$\begin{aligned}\mathbb{E}\left[\mathbf{R}_t\right] &= S\sum_{e^*\in A^*}\bar{w}(e^*) - (S-1)\sum_{e'\in\mathbf{B}_t}\bar{w}(e') - \sum_{e\in\mathbf{D}_t}\bar{w}(e)\\ &= S\left(\sum_{e^*\in A^*}\bar{w}(e^*) - \sum_{e\in\mathbf{D}_t}\bar{w}(e)\right) + \\ &\quad (S-1)\left(\sum_{e\in\mathbf{D}_t}\bar{w}(e) - \sum_{e'\in\mathbf{B}_t}\bar{w}(e')\right) ,\end{aligned} \tag{5.10}$$

where the first term represents regret due to the decision set and the second term represents regret due to interleaving with the baseline set.

The first term in (5.10) can be bounded as follows. Since the decision set $\mathbf{D}_t$ is chosen optimistically, the UCBs of items in $\mathbf{D}_t$ are at least as high as those of the matched items in A^*, and we have that

$$\sum_{e^* \in A^*} \bar{w}(e^*) - \sum_{e \in \mathbf{D}_t} \bar{w}(e) \leqslant \sum_{e \in \mathbf{D}_t} 2c_{n,\mathbf{T}_{t-1}(e)} \tag{5.11}$$

at any time t under event $\mathcal{E}$, from Lemma 5.8 in Appendix. Now we add all above upper bounds over time and get

$$\begin{aligned} &\sum_{t=1}^{n/S} \sum_{e \in \mathbf{D}_t} 2c_{n,\mathbf{T}_{t-1}(e)} \\ &\quad \leqslant \sum_{e \in E \setminus A^*} \sum_{t=1}^{n/S} \sqrt{\frac{6 \log n}{\mathbf{T}_{t-1}(e)}} \mathbb{1}(e \in \mathbf{D}_t) \\ &\quad \leqslant \sum_{e \in E \setminus A^*} \sqrt{6 \log n} \left(1 + 2\sqrt{\frac{6 \log n}{\Delta_{e,\min}^2}}\right) \\ &\quad = \sum_{e \in E \setminus A^*} \frac{12}{\Delta_{e,\min}} \log n + L\sqrt{6 \log n}\,. \end{aligned} \tag{5.12}$$

The first inequality is from the definition of our confidence intervals. The second inequality is from two observations. First, the counter $\mathbf{T}_t(e)$ increases whenever item e is chosen. Second, this event occurs at most $m = 6\Delta_{e,\min}^{-2} \log n$ times (Lemma 5.8 in Appendix). Finally, we apply

$$\sum_{s=1}^{m} \frac{1}{\sqrt{s}} \leqslant 1 + 2\sqrt{m}\,. \tag{5.13}$$

The last equality is an algebraic manipulation.

The second term in (5.10) is bounded as follows. Since the baseline set $\mathbf{B}_t$ is chosen based on LCBs, the LCBs of items in $\mathbf{B}_t$ are at least as high as those of the matched items in $\mathbf{D}_t$, and we have that

$$\sum_{e \in \mathbf{D}_t} \bar{w}(e) - \sum_{e' \in \mathbf{B}_t} \bar{w}(e') \leqslant \sum_{e \in \mathbf{D}_t} 2c_{n,\mathbf{T}_{t-1}(e)} \tag{5.14}$$

at any time t under event $\mathcal{E}$, from Lemma 5.9 in Appendix. Now we add all above upper bounds over time and get

$$\begin{aligned}
\sum_{t=1}^{n/S} \sum_{e \in \mathbf{D}_t} 2c_{n,\mathbf{T}_{t-1}(e)} \leqslant \\
\sum_{e \in E \setminus A^*} \sum_{t=1}^{n/S} \sqrt{\frac{6 \log n}{\mathbf{T}_{t-1}(e)}} \mathbb{1}(e \in \mathbf{D}_t) + \\
\sum_{e^* \in A^*} \sum_{t=1}^{n/S} \sqrt{\frac{6 \log n}{\mathbf{T}_{t-1}(e^*)}} \mathbb{1}(e^* \in \mathbf{D}_t) \ ,
\end{aligned} \tag{5.15}$$

where the inequality is from the definition of our confidence intervals.

The first term in (5.15) is bounded as in (5.12). The second term is bounded similarly, where the only difference is in the definition of the gap. In particular, if an optimal item e^* is chosen $\Omega((\Delta^*_{e^*,\min})^{-2} \log n)$ times (Lemma 5.9 in Appendix), it must be in the baseline set $\mathbf{B}_t$ and the corresponding regret is zero. Therefore, the regret in (5.15) is bounded from above by

$$\sum_{e \in E \setminus A^*} \frac{12}{\Delta_{e,\min}} \log n + L\sqrt{6 \log n} + \sum_{e^* \in A^*} \frac{12}{\Delta^*_{e^*,\min}} \log n + L\sqrt{6 \log n} \, . \tag{5.16}$$

Finally, we add S times the upper bound in (5.12) and $S - 1$ times the upper bound in (5.16), and get our claim. □

5.4.2 iUCB2: Unknown Baseline Mean Rewards

First, we show that iUCB2 is conservative. The proof of this claim is in Section 5.8.2.

Theorem 5.4. *iUCB2 satisfies* (5.1) *jointly at all times* $t \in [n]$ *with probability of at least* $1 - 2L/(Kn)$.

Now we prove a gap-dependent upper bound on the regret of iUCB2.

Theorem 5.5 (Regret of `iUCB2`). *The expected* n*-step regret of* `iUCB2` *is bounded as*

$$(S-1)\left(\sum_{e\in E\setminus A^*}\frac{48}{\Delta_{e,\min}}+\sum_{e^*\in A^*}\frac{36}{\Delta^*_{e^*,\min}}\right)\log n+\sum_{e\in E\setminus A^*}\frac{12}{\Delta_{e,\min}}\log n+c\,,$$

where $S = 1/\alpha$*;* $\Delta_{e,\min}$ *and* $\Delta^*_{e^*,\min}$ *are defined in* (5.7) *and* (5.8)*, respectively; and* $c = O(SL\sqrt{\log n})$*.*

Proof. The proof is similar to that of Theorem 5.3. The only major difference is that items in the default baseline action B_0 are chosen to $\mathbf{B}_t$ based on their UCBs, while the other items are selected based on their LCBs.

The regret at time t decomposes as in (5.10), and the first term in (5.10) is bounded exactly as in (5.12). To bound the second term, we decompose the regret based on whether the item in $\mathbf{B}_t$ is in B_0 or not, and get that

$$\begin{aligned}
&\sum_{e\in\mathbf{D}_t}\bar{w}(e)-\sum_{e'\in\mathbf{B}_t}\bar{w}(e') \\
&\quad=\sum_{e\in\mathbf{D}_t:\mathbf{œ}_t(e)\notin B_0}\bar{w}(e)-\sum_{e'\in\mathbf{B}_t\setminus B_0}\bar{w}(e')+\sum_{e\in\mathbf{D}_t:\mathbf{œ}_t(e)\in B_0}\bar{w}(e)-\sum_{e'\in\mathbf{B}_t\cap B_0}\bar{w}(e') \\
&\quad\leqslant\sum_{e\in\mathbf{D}_t:\mathbf{œ}_t(e)\notin B_0}2c_{n,\mathbf{T}_{t-1}(e)}+\sum_{e\in\mathbf{D}_t:\mathbf{œ}_t(e)\in B_0}4c_{n,\mathbf{T}_{t-1}(e)}\,,
\end{aligned}$$

where $\mathbf{œ}_t(e)$ is the matched item in $\mathbf{B}_t$ for item e in $\mathbf{D}_t$. The last step follows from two observations. When $\mathbf{œ}_t(e)\notin B_0$, we follow the same proof as in (5.14) and get the same upper bound as in (5.16). When $\mathbf{œ}_t(e)\in B_0$, we apply Lemma 5.10 in Appendix. This lemma relies on the observation that any item in $\mathbf{B}_t\cap B_0$ is chosen at least as often as its matched item in $\mathbf{D}_t$ up to any time t, which holds for any $\alpha\leqslant 1/2$. The final upper bound is the same as in (5.16), except that all terms are multiplied by 2.

Finally, we add up the contributions of all terms, which is S times the upper bound in (5.12) and $3(S-1)$ times the upper bound in (5.16), and get our claim. □

5.4.3 Discussion

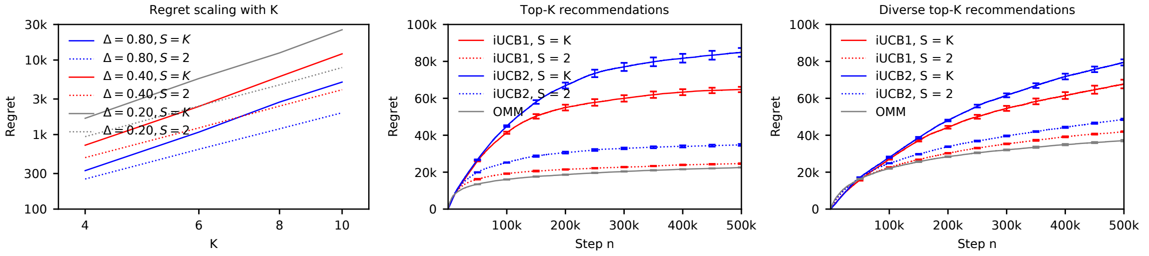
Our regret bounds in Theorems 5.3 and 5.5 depend on two gaps. The first gap, $\Delta_{e,\min}$ in (5.7), measures the distance of suboptimal item e from the closest better optimal item. This gap is standard in stochastic combinatorial semi-bandits with matroid constraints [Kveton et al., 2014b], which we refer to as *matroid bandits*. Matroid constraints are a weaker notion of exchangeability than that in this chapter. The second gap, $\Delta^*_{e^*,\min}$ in (5.8), measures the distance of optimal item e^* from the closest worse suboptimal item. Similar gaps appear in top-K best-arm identification problems [Kalyanakrishnan et al., 2012]. If we let

$$\Delta = \min\{\min_{e \in E \setminus A^*} \Delta_{e,\min}, \min_{e^* \in A^*} \Delta^*_{e^*,\min}\},$$

then the bounds in Theorems 5.3 and 5.5 scale as $O(SL\Delta^{-1})$, where L is the number of items and $S = 1/\alpha$ is the number of interleaved actions in iUCB to observe each item in the decision set once. We validate this scaling empirically in Section 5.5.1.

When compared to matroid bandits [Kveton et al., 2014b, Talebi and Proutiere, 2016], our regret bounds contain an extra factor of S. This is the *price of conservativism*. In particular, since iUCB takes S interleaved actions to observe each item in the decision set $\mathbf{D}_t$ once, its regret is S times higher than that of the algorithm that can explore $\mathbf{D}_t$ in a single action. Note that whenever $\alpha = \Omega(1)$, as at $\alpha = 1/2$, the extra factor of $S = 1/\alpha$ is a constant independent of K and our bounds scale as those in matroid bandits [Kveton et al., 2014b, Talebi and Proutiere, 2016].

Finally, by a standard gap-dependent to gap-free reduction, where the gaps are divided in into those that are larger than ε and smaller than ε, and then ε is tuned, we have a gap-free regret bound of $O(S\sqrt{KLn \log n})$. This bound is again at most S times higher than that in matroid bandits [Kveton et al., 2014b].



(a) (b) (c)

Figure 5.1: **a**. The n-step regret of `iUCB1` in the synthetic problem in Section 5.5.1 as a function of K. **b**. The regret of `iUCB1`, `iUCB2`, and `OMM` in the top-K recommendation problem in Section 5.5.2. **c**. The regret of `iUCB1`, `iUCB2`, and `OMM` in the diverse top-K recommendation problem in Section 5.5.2.

5.5 Experiments

We conduct two experiments. In Section 5.5.1, we validate that the regret of `iUCB1` grows as suggested by our upper bound in Theorem 5.3. In Section 5.5.2, we apply `iUCB` to two recommendation problems. We also compare it to a non-conservative algorithm `OMM` [Kveton et al., 2014b], which can learn optimal actions in our problems; but also severely violates the conservative constraint in (5.1).

5.5.1 Regret Scaling

The first experiment validates that the regret of `iUCB1` scales as suggested by our gap-dependent upper bound in Theorem 5.3. The ground set is $E = [K^2]$ for parameter $K > 0$ and the action set is $\mathcal{B} = \Pi_K(E)$. The i-th entry of weight vector $\mathbf{w}_t$, $\mathbf{w}_t(i)$, is an independent Bernoulli random variable with mean

$$\bar{w}(i) = 0.5(1 - \Delta \mathbb{1}(i > K))$$

for $\Delta \in (0, 1)$. From the definition of $\bar{w}$, the optimal action is $A^* = [K]$. The default baseline action are the last K items in E, $B_0 = [K^2] \setminus [K(K-1)]$. In this problem, we expect the regret of `iUCB1` to scale as $SK^2\Delta^{-1}$.

We vary K, Δ, and S; and report the n-step regret of `iUCB1` in 100k steps in

Fig. 5.1a. The regret is shown in log-log plots as a function of K for three values of Δ and two values of S. We observe two major trends. First, the regret grows as S and K increase, and Δ decreases. This is consistent with our theoretical analysis. Second, the growth rate is as predicted. In particular, when $S = K$, and one decision item is interleaved with $K - 1$ baseline items, the slopes of the plots are close to 3. This confirms cubic dependence on K when $S = K$. Moreover, when $S = 2$, and $K/2$ decision items are interleaved with $K/2$ baseline items, the slopes of the plots are close to 2. This confirms quadratic dependence on K when $S = 2$.

5.5.2 Recommender System Experiment

In the second experiment, we apply `iUCB` to two motivating recommendation problems in Section 5.2.4. In both problems, we recommend K movies out of L. The attraction of movies is estimated from the *MovieLens* dataset from February 2003 [Lam and Herlocker, 2013], where 6 thousand users give one million ratings to 4 thousand movies.

Our learning problems are formulated as follows. The set E are 200 movies from the MovieLens dataset. The set is partitioned as $E = \bigcup_{i=1}^{10} E_i$, where E_i are 20 most popular movies in the i-th most popular MovieLens movie genre that are not in $E_1, \dots, E_{i-1}$. The weight of item e at time t, $\mathbf{w}_t(e) \in \{0, 1\}$, indicates that item e attracts the user at time t. We set it as $\mathbf{w}_t(e) = 1$ if and only if the user rated item e in our dataset. This indicates that the user watched movie e before, perhaps because the movie was attractive. The user at time t is drawn randomly from all MovieLens users. The objective of the learning agent is to learn a set of items with the highest expected attraction over all users.

We study two recommendation problems. The first problem is *top-K recommendation* in Section 5.2.4, where $K = 10$. The exchangeable action set is $\mathcal{B} = \Pi_K(E)$, all sets of size K from E. The optimal action A^* are 10 most attractive movies. The default baseline action B_0 are the next 10 most attractive movies. We choose B_0 in this way because existing baseline policies tend to perform well.

The second problem is *diverse top-K recommendation* in Section 5.2.4, where $K = 10$.

The exchangeable action set is defined as in (5.3), where each $\mathcal{P}_i$ is associated with movie group E_i. The optimal action A^* is the set of most attractive movies from all E_i. The default baseline action B_0 is the set of second most attractive movies from all E_i. Again, we choose B_0 in this way because existing baseline policies tend to perform well.

Our results are reported in Figures 5.1b and 5.1c. We observe several trends across both problems. First, the regret of all algorithms is concave, which shows that they learn better policies over time. Second, the regret of `iUCB2` is higher than that of `iUCB1`. This is because `iUCB2` does not know the values of default baseline items B_0, while `iUCB1` does. Since `iUCB2` has to estimate these values, it is more conservative and learns slower. Second, the regret increases with S. For instance, in Fig. 5.1b, the regret at $S = K$ is almost twice as high as that at $S = 2$. This is expected since the former setting is more conservative. In particular, at $S = K$, one decision item is interleaved with $K - 1$ baseline items; while at $S = 2$, and $K/2$ decision items are interleaved with $K/2$ baseline items.

Finally, we note that `OMM` achieves the lowest regret. But it also violates our conservative constraints. For instance, at $S = K$, `iUCB1` and `iUCB2` violate none of the constraints in (5.1). On the other hand, `OMM` violates more than 16k and 158k constraints in Figures 5.1b and 5.1c, respectively, on average in 500k steps. This is one violated constraint in every three actions in the latter problem. We also note that at $S = 2$, the regret of `iUCB1` approaches that of `OMM`. This indicates that reasonably conservative constraints, such as that one half of the recommended items are at least as good as default baseline items in B_0, can be satisfied without major impact on regret.

5.6 Related Work

The idea of controlled exploration in multi-armed bandits is not new. Wu et al. [2016] studied conservatism in multi-armed bandits, where the learning agent is constrained to have its *cumulative* reward no worse than $1 - \alpha$ fraction of that of the default action. In our setting, the cumulative nature of this constraint means that the agent can take disastrous actions, with many suboptimal items, once in every $1/\alpha$ steps. Our *per-time* constraint in (5.1) explicitly prohibits this design and such disastrous actions. However, note that our setting and algorithms apply only to combinatorial action spaces, and hence are less general.

A/B testing [Siroker and Koomen, 2013] can also solve constrained exploration problems. When the new and default actions are chosen randomly with probabilities α and $1-\alpha$, respectively, the *expected* reward is no worse than $1-\alpha$ fraction of that of the default action. Since this constraint is *in expectation*, A/B testing can take disastrous actions occasionally. In comparison, we satisfy our constraint in (5.1) with a *high-probability* at all times, and strictly avoid disastrous actions.

Online learning with matroids was introduced by Kveton et al. [2014b] and later studied by Talebi and Proutiere [2016]. These works do not consider any notion of conservatism. A naive generalization of these works to conservatism is problematic, as discussed at the beginning in Section 5.3.

Kazerouni et al. [2017] studied conservatism in linear bandits. Similarly to Wu et al. [2016], their constraint is *cumulative*. Furthermore, the time complexity of their algorithm increases with time when the expected reward of the baseline policy is unknown. In comparison, `iUCB` is both computationally and sample efficient.

Bastani et al. [2017] studied contextual bandits and proposed diversity assumptions on the environment. Intuitively, if the context varies a lot over time, the environment explores on behalf of the learning agent, and the agent does not have to explore. In comparison, we actively explore in a constrained fashion.

Radlinski and Joachims [2006] proposed randomizing the order of presented items to estimate their relevance in the presence of item and position biases. Their algorithm guarantees that the quality of the presented items is affected minimally. But it does not learn a better policy. The idea of interleaving has been used to evaluate information retrieval systems and Chapelle et al. [2012] validated its efficacy. Chapelle et al. [2012] did not study the problem of learning a better policy. `iUCB` learns a better policy. While we do not consider item and position biases in this work, we hope to do so in future work.

5.7 Conclusions

In this chapter, we study controlled exploration in combinatorial action spaces using interleaving, and precisely formulate the learning problem in the space of exchangeable actions. Our conservative formulation is more suitable for combinatorial spaces than existing notions of conservatism. We propose an algorithm for solving our problem, iUCB, and prove gap-dependent upper bounds on its regret. iUCB exploits the idea of interleaving and can evaluate a disastrous action without ever taking it.

We leave open several questions of interest. First, how large is the class of exchangeable action spaces? We provide two examples of such spaces in Section 5.2.3 in relation to top-K and diverse top-K recommendations. A fairly large class of exchangeable action spaces is the class of *strongly base-orderable matroids*. The action spaces in top-K and diverse top-K recommendation problems belong to this class.

Second, in general it may not be possible to build unbiased estimators of item relevances with interleaving, as clicks are typically biased due to the position of the item and other recommended items [Chuklin et al., 2015c]. Nevertheless, we believe that it is possible to build biased estimators with the right bias, such that a more relevant item never appears to be less relevant than a less relevant item [Zoghi et al., 2017]. We leave this for future work.

Third, we not only require the action space to be exchangeable, but also need to construct the bijection in Definition 5.1. The construction is straightforward for uniform and partition matroids in our experiments.

We also leave open the question of a lower bound. Finally, we wish to highlight that new ideas in our analysis of iUCB can be used to greatly simplify the original analysis of OMM in Kveton et al. [2014b].

5.8 Appendix

Lemma 5.6. *Let* $\mathcal{E}_r$ *be the good event in* (5.6)*. Then*

$$\mathbb{P}\left(\bigcup_{r=1}^{n/K} \bar{\mathcal{E}}_r\right) \leqslant \sum_{r=1}^{n/K} \mathbb{E}\,\square\,\mathbb{1}\left(\bar{\mathcal{E}}_r\right) \leqslant \frac{2L}{Kn}\,.$$

Proof. From the definition of our confidence intervals and Hoeffding's inequality [Boucheron et al., 2013],

$$\mathbb{P}(|\bar{w}(e) - \hat{\mathbf{w}}_s(e)| \geqslant c_{t,s}) \leqslant 2\exp[-3\log t]$$

for any $e \in E$, $s \in [n]$, and $t \in [n]$. Therefore,

$$\begin{aligned}
\mathbb{P}\left(\bigcup_{r=1}^{n/K} \bar{\mathcal{E}}_r\right) &\leqslant \sum_{r=1}^{n/K} \mathbb{P}(\bar{\mathcal{E}}_r) \\
&\leqslant \sum_{r=1}^{n/K} \sum_{e \in E} \sum_{s=1}^{rK} \mathbb{P}(|\bar{w}(e) - \hat{\mathbf{w}}_s(e)| \geqslant c_{n,s}) \\
&\leqslant 2 \sum_{e \in E} \frac{1}{Kn}\,.
\end{aligned}$$

This concludes our proof. □

Lemma 5.7. *Let* A *be the maximum weight action with respect to weights* w*. Let* B *be any action and let* $\rho : A \to B$ *be the bijection in Definition 5.1. Then*

$$\forall a \in A : w(a) \geqslant w(\rho(a))\,.$$

Proof. Fix $a \in A$ and let $b = \rho(a)$. By Definition 5.1, $A_b^a = A \setminus \{a\} \cup \{b\} \in \mathcal{B}$. Now note that A is the maximum weight action with respect to w. Therefore,

$$w(a) - w(b) = \sum_{e \in A} w(e) - \sum_{e \in A_b^a} w(e) \geqslant 0\,.$$

This concludes our proof. □

5.8.1 `iUCB1`: Known Baseline Means

Theorem 5.2. *iUCB1 satisfies* (5.1) *jointly at all times* $t \in [n]$ *with probability of at least* $1 - 2L/(Kn)$.

Proof. At time t, the baseline set $\mathbf{B}_t$ is the maximum weight action with respect to $\mathbf{v}_t$. Therefore, by Lemma 5.7, there exists a bijection $\boldsymbol{æ} : \mathbf{B}_t \rightarrow B_0$ such that

$$\forall b \in \mathbf{B}_t : \mathbf{v}_t(b) \geqslant \mathbf{v}_t(\boldsymbol{æ}(b))\,.$$

From the definition of $\mathbf{v}_t$, $\mathbf{v}_t(\boldsymbol{æ}(b)) = \bar{w}(\boldsymbol{æ}(b))$ for any $b \in \mathbf{B}_t$, and thus

$$\forall b \in \mathbf{B}_t : \mathbf{v}_t(b) \geqslant \bar{w}(\boldsymbol{æ}(b))\,.$$

Now suppose that event $\mathcal{E}_t$ in (5.6) happens. Then $\bar{w}(e) \geqslant \mathbf{L}_t(e)$ for any $e \in E$, and it follows that

$$\forall b \in \mathbf{B}_t : \bar{w}(b) \geqslant \bar{w}(\boldsymbol{æ}(b))\,.$$

Since any action at time t contains $K(1-\alpha)$ items from $\mathbf{B}_t$, the constraint in (5.1) is satisfied when event $\mathcal{E}_t$ happens.

Finally, we prove that $\mathbb{P}(\cup_t \bar{\mathcal{E}}_t) \leqslant 2L/(Kn)$ in Lemma 5.6. Therefore, $\mathbb{P}(\mathcal{E}_t) \geqslant \mathbb{P}(\cap_t \mathcal{E}_t) \geqslant 1 - 2L/(Kn)$. This concludes our proof. □

Lemma 5.8. *For any* $e^* \in A^*$, $e \in \mathbf{D}_t$ *such that* $e = \mathbf{ß}_t(e^*)$, *we have that*

$$\Delta_{e,e^*} \leqslant 2c_{n,\mathbf{T}_{t-1}(e)}, \qquad \textit{and} \qquad \mathbf{T}_{t-1}(e) \leqslant \frac{6\log n}{\Delta^2_{e,e^*}} \leqslant \frac{6\log n}{\Delta^2_{e,\min}}, \tag{5.17}$$

where $\Delta_{e,\min}$ *is defined in* (5.7).

Proof. Since the decision set $\mathbf{D}_t$ is chosen using upper confidence bounds, we have that $\mathbf{U}_t(e) \geqslant \mathbf{U}_t(e^*)$. This gives us:

$$\bar{w}(e) + 2c_{n,\mathbf{T}_{t-1}(e)} \geqslant \hat{\mathbf{w}}_{t-1}(e) + c_{n,\mathbf{T}_{t-1}(e)} = \mathbf{U}_t(e) \geqslant \mathbf{U}_t(e^*) \geqslant \bar{w}(e^*).$$

This implies the first inequality in (5.17). Substituting the expression for $c_{n,\mathbf{T}_{t-1}(e)}$ from (5.5) yields the bound on $\mathbf{T}_{t-1}(e)$ in (5.17). □

Lemma 5.9. *For any* $e^* \in A^*$, $e \in \mathbf{D}_t$, *and* $e' \in \mathbf{B}_t$ *such that* $e = \mathbf{ß}_t(e^*)$ *and* $e' = \mathbf{œ}_t(e)$,

$$\Delta_{e',e} \leqslant 2c_{n,\mathbf{T}_{t-1}(e)}.$$

Furthermore, if $e \in A^*$, *then* $e = e^*$ *and*

$$2c_{n,\mathbf{T}_{t-1}(e^*)} \geqslant \bar{w}(e^*) - \bar{w}(e'), \qquad \textit{and} \qquad \mathbf{T}_{t-1}(e^*) \leqslant \frac{6\log n}{\Delta^2_{e',e^*}} \leqslant \frac{6\log n}{\Delta^{*2}_{e^*,\min}}, \tag{5.18}$$

where $\Delta^*_{e^*,\min}$ *is defined in* (5.8).

Proof. Since the baseline set is selected using lower confidence bounds, we have that $\mathbf{L}_t(e') \geqslant \mathbf{L}_t(e)$. This gives us:

$$\bar{w}(e') \geqslant \mathbf{L}_t(e') \geqslant \mathbf{L}_t(e) \geqslant \bar{w}(e) - 2c_{n,\mathbf{T}_{t-1}(e)}$$

This implies that

$$2c_{n,\mathbf{T}_{t-1}(e)} \geqslant \bar{w}(e) - \bar{w}(e'). \tag{5.19}$$

If $e \in A^*$, then since $e = \textbf{ß}_t(e^*)$, we must have that $e = e^*$. Assume otherwise. Then $A^* \setminus \{e^*\} \cup \{e\}$ is a action (by Definition 5.1) of size $(K-1)$, which contradicts the fact that all actions have the same cardinality K. Substituting $e = e^*$ in (5.19) gives the first inequality in (5.18). The $\mathbf{T}_{t-1}(e^*)$ bound in (5.18) follows by substituting the expression of $c_{n,\mathbf{T}_{t-1}(e)}$ from (5.5). □

5.8.2 `iUCB2`: Unknown Baseline Means

Theorem 5.4. *iUCB2 satisfies* (5.1) *jointly at all times* $t \in [n]$ *with probability of at least* $1 - 2L/(Kn)$.

Proof. At time t, the baseline set $\mathbf{B}_t$ is the maximum weight action with respect to $\mathbf{v}_t$. Therefore, by Lemma 5.7, there exists a bijection $\textbf{æ} : \mathbf{B}_t \to B_0$ such that

$$\forall b \in \mathbf{B}_t : \mathbf{v}_t(b) \geqslant \mathbf{v}_t(\textbf{æ}(b))\,.$$

Now we consider two cases. First, suppose that $b \in B_0$. Then by Lemma 5.7, $b = \textbf{æ}(b)$, and $\bar{w}(b) \geqslant \bar{w}(\textbf{æ}(b))$ from our assumption. Second, suppose that $b \notin B_0$. Then from $\mathbf{v}_t(b) = \mathbf{L}_t(b)$ and $\mathbf{v}_t(\textbf{æ}(b)) = \mathbf{U}_t(\textbf{æ}(b))$, and

$$\bar{w}(b) \geqslant \mathbf{L}_t(b) \geqslant \mathbf{U}_t(\textbf{æ}(b)) \geqslant \bar{w}(\textbf{æ}(b))$$

under event $\mathcal{E}_t$. Since any action at time t contains $K(1-\alpha)$ items from $\mathbf{B}_t$, the constraint in (5.1) is satisfied when event $\mathcal{E}_t$ happens.

Finally, we prove that $\mathbb{P}(\cup_t \bar{\mathcal{E}}_t) \leqslant 2L/(Kn)$ in Lemma 5.6. Therefore, $\mathbb{P}(\mathcal{E}_t) \geqslant \mathbb{P}(\cap_t \mathcal{E}_t) \geqslant 1 - 2L/(Kn)$. This concludes our proof. □

Lemma 5.10. *For any* $e^* \in A^*, e \in \mathbf{D}_t$, *and* $e' \in \mathbf{B}_t$ *such that* $e' \in B_0, e = \textbf{ß}_t(e^*)$, *and* $e' \in \textbf{œ}_t(e)$,

$$\Delta_{e',e} \leqslant 4c_{n,\mathbf{T}_{t-1}(e)}.$$

Proof. For items $e' \in B_0 \cap \mathbf{B}_t$, we have that $\mathbf{U}_t(e') \geqslant \mathbf{L}_t(e)$. This gives us

$$\bar{w}(e') + 2c_{n,\mathbf{T}_{t-1}(e')} \geqslant \mathbf{U}_t(e') \geqslant \mathbf{L}_t(e) \geqslant \bar{w}(e) - 2c_{n,\mathbf{T}_{t-1}(e)}$$

This implies that

$$\Delta_{e',e} \leqslant 2c_{n,\mathbf{T}_{t-1}(e)} + 2c_{n,\mathbf{T}_{t-1}(e')}. \tag{5.20}$$

An item eliminated from the baseline set $\mathbf{B}_t$ is never re-introduced in the baseline set. Since $e' \in B_0 \cap \mathbf{B}_t$, it must have never been eliminated from the baseline set. The maximum number of times e can be played is by including it in every decision set $\mathbf{D}_t$. In any round, since the baseline items are played $(S-1)$ times the decision set counterparts, and $S \geqslant 2$, we have that $\mathbf{T}_{t-1}(e') \geqslant (S-1)\mathbf{T}_{t-1}(e) \geqslant \mathbf{T}_{t-1}(e)$, which implies that

$$c_{n,\mathbf{T}_{t-1}(e')} \leqslant c_{n,\mathbf{T}_{t-1}(e)}.$$

Substituting in (5.20),

$$\Delta_{e',e} \leqslant 4c_{n,\mathbf{T}_{t-1}(e)}.$$

□

Algorithm 5 `iUCB` for conservative interleaving bandits.

```
Input: Baseline action $B_0 \in \mathcal{B}$, risk tolerance $\alpha$

$S \leftarrow 1/\alpha \in \mathbb{N}$
Observe $\mathbf{w}_0 \sim P$
$\forall e \in E : \mathbf{T}_0(e) \leftarrow 1,\ \hat{\mathbf{w}}_1(e) \leftarrow \mathbf{w}_0(e)$

for $t = 1, 2, \ldots$ do
    for all $e \in E$ do {                          // Compute UCBs and LCBs}
        $\mathbf{U}_t(e) = \hat{\mathbf{w}}_{\mathbf{T}_{t-1}(e)}(e) + c_{n,\mathbf{T}_{t-1}(e)}$
        $\mathbf{L}_t(e) = \max\{\hat{\mathbf{w}}_{\mathbf{T}_{t-1}(e)}(e) - c_{n,\mathbf{T}_{t-1}(e)}, 0\}$
    end for

    $\mathbf{D}_t \leftarrow \mathsf{OPT}(\mathbf{U}_t)$                          // Compute decision set
    for all $e \in B_0$ do {                           // Compute baseline set}
        if $\bar{w}(e)$ is known then {                          // iUCB1}
            $\mathbf{v}_t(e) \leftarrow \bar{w}(e)$
        else {                                              // iUCB2}
            $\mathbf{v}_t(e) \leftarrow \mathbf{U}_t(e)$
        end if
    end for
    for all $e \in E \setminus B_0$ do
        $\mathbf{v}_t(e) \leftarrow \mathbf{L}_t(e)$
    end for
    $\mathbf{B}_t \leftarrow \mathsf{OPT}(\mathbf{v}_t)$

    // Take S combined actions and update statistics
    Let $\{\mathbf{B}_t^s\}_{s=1}^S$ be a partition of $\mathbf{B}_t$ such that $|\mathbf{B}_t^s| = \alpha K$ for all $s \in [S]$
    Let $\mathbf{æ}_t : \mathbf{B}_t \to \mathbf{D}_t$ be the bijection in Definition 5.1
    $\forall e \in E : \mathbf{T}_t(e) \leftarrow \mathbf{T}_{t-1}(e)$
    for $s = 1, \ldots, S$ do
        Take action $\mathbf{A}_t = \mathbf{B}_t \setminus \mathbf{B}_t^s \cup \{\mathbf{æ}_t(e) : e \in \mathbf{B}_t^s\}$
        Observe $\{\mathbf{w}_t(e) : e \in \mathbf{A}_t\}$, where $\mathbf{w}_t \sim P$
        for all $e \in \mathbf{A}_t$ do
            $\hat{\mathbf{w}}_{\mathbf{T}_t(e)+1}(e) \leftarrow \frac{\mathbf{T}_t(e)\hat{\mathbf{w}}_{\mathbf{T}_t(e)}(e) + \mathbf{w}_t(e)}{\mathbf{T}_t(e) + 1}$
            $\mathbf{T}_t(e) \leftarrow \mathbf{T}_t(e) + 1$
        end for
    end for
end for
```

6 STOCHASTIC RANK-1 BANDITS

6.1 Introduction

We study the problem of finding the maximum entry of a stochastic rank-1 matrix from noisy and adaptively-chosen observations. This problem is motivated by two problems, ranking in the position-based model Richardson et al. [2007] and online advertising.

The *position-based model (PBM)* Richardson et al. [2007] is one of the most fundamental click models Chuklin et al. [2015b], a model of how people click on a list of K items out of L. This model is defined as follows. Each *item* is associated with its *attraction* and each *position* in the list is associated with its *examination*. The attraction of any item and the examination of any position are i.i.d. Bernoulli random variables. The item in the list is *clicked* only if it is attractive and its position is examined. Under these assumptions, the pair of the item and position that maximizes the probability of clicking is the maximum entry of a rank-1 matrix, which is the outer product of the attraction probabilities of items and the examination probabilities of positions.

As another example, consider a marketer of a product who has two sets of actions, K population *segments* and L marketing *channels*. Given a product, some segments are *easier to market to* and some channels are *more appropriate*. Now suppose that the conversion happens only if both actions are successful and that the successes of these actions are independent. Then similarly to our earlier example, the pair of the population segment and marketing channel that maximizes the conversion rate is the maximum entry of a rank-1 matrix.

We propose an online learning model for solving our motivating problems, which we call a *stochastic rank-*1 *bandit*. The learning agent interacts with our problem as follows. At time t, the agent selects a pair of row and column arms, and receives the product of their individual values as a reward. The values are stochastic, drawn independently, and not observed. The goal of the agent is to maximize its expected cumulative reward, or equivalently to minimize its expected cumulative regret with respect to the optimal solution, the most rewarding pair of row and column arms.

We make five contributions. First, we precisely formulate the online learning problem of *stochastic rank*-1 *bandits*. Second, we design an elimination algorithm for solving it, which we call `Rank1Elim`. The key idea in `Rank1Elim` is to explore all remaining rows and columns randomly over all remaining columns and rows, respectively, to estimate their expected rewards; and then eliminate those rows and columns that seem suboptimal. This algorithm is computationally efficient and easy to implement. Third, we derive a $O((K + L)(1/\Delta) \log n)$ gap-dependent upper bound on its n-step regret, where K is the number of rows, L is the number of columns, and Δ is the minimum of the row and column gaps; under the assumption that the mean row and column rewards are bounded away from zero. Fourth, we derive a nearly matching gap-dependent lower bound. Finally, we evaluate our algorithm empirically. In particular, we validate the scaling of its regret, compare it to multiple baselines, and show that it can learn near-optimal solutions even if our modeling assumptions are mildly violated.

We denote random variables by boldface letters and define $[n] = \{1, \dots, n\}$. For any sets A and B, we denote by A^B the set of all vectors whose entries are indexed by B and take values from A.

6.2 Setting

We formulate our online learning problem as a *stochastic rank*-1 *bandit*. An instance of this problem is defined by a tuple (K, L, P_U, P_V), where K is the number of rows, L is the number of columns, P_U is a probability distribution over a unit hypercube $[0, 1]^K$, and P_V is a probability distribution over a unit hypercube $[0, 1]^L$.

Let $(\mathbf{u}_t)_{t=1}^n$ be an i.i.d. sequence of n vectors drawn from distribution P_U and $(\mathbf{v}_t)_{t=1}^n$ be an i.i.d. sequence of n vectors drawn from distribution P_V, such that $\mathbf{u}_t$ and $\mathbf{v}_t$ are drawn independently at any time t. The learning agent interacts with our problem as follows. At time t, it chooses *arm* $(\mathbf{i}_t, \mathbf{j}_t) \in [K] \times [L]$ based on its history up to time t; and then *observes* $\mathbf{u}_t(\mathbf{i}_t)\mathbf{v}_t(\mathbf{j}_t)$, which is also its *reward*.

The goal of the agent is to maximize its expected cumulative reward in n steps. This is equivalent to minimizing the *expected cumulative regret* in n steps

$$R(n) = \mathbb{E}\left[\sum_{t=1}^{n} R(\mathbf{i}_t, \mathbf{j}_t, \mathbf{u}_t, \mathbf{v}_t)\right],$$

where $R(\mathbf{i}_t, \mathbf{j}_t, \mathbf{u}_t, \mathbf{v}_t) = \mathbf{u}_t(i^*)\mathbf{v}_t(j^*) - \mathbf{u}_t(\mathbf{i}_t)\mathbf{v}_t(\mathbf{j}_t)$ is the *instantaneous stochastic regret* of the agent at time t and

$$(i^*, j^*) = \underset{(i,j)\in[K]\times[L]}{\arg\max}\ \mathbb{E}\left[\mathbf{u}_1(i)\mathbf{v}_1(j)\right]$$

is the *optimal solution* in hindsight of knowing P_U and P_V. Since $\mathbf{u}_1$ and $\mathbf{v}_1$ are drawn independently, and $\mathbf{u}_1(i) \geqslant 0$ for all $i \in [K]$ and $\mathbf{v}_1(j) \geqslant 0$ for all $j \in [L]$, we get that

$$i^* = \underset{i\in[K]}{\arg\max}\ \mu\bar{u}(i), \quad j^* = \underset{j\in[L]}{\arg\max}\ \mu\bar{v}(j),$$

for any $\mu > 0$, where $\bar{u} = \mathbb{E}[\mathbf{u}_1]$ and $\bar{v} = \mathbb{E}[\mathbf{v}_1]$. This is the key idea in our solution.

Note that the problem of learning $\bar{u}$ and $\bar{v}$ from stochastic observations $\{\mathbf{u}_t(\mathbf{i}_t)\mathbf{v}_t(\mathbf{j}_t)\}_{t=1}^{n}$ is a special case of *matrix completion from noisy observations* Keshavan et al. [2010]. This problem is harder than that of learning (i^*, j^*). In particular, the most popular approach to matrix completion is alternating minimization of a non-convex function Koren et al. [2009], where the observations are corrupted with Gaussian noise. In contrast, our proposed algorithm is guaranteed to learn the optimal solution with a high probability, and does not make any strong assumptions on P_U and P_V.

6.3 Naive Solutions

Our learning problem is a KL-arm bandit with $K + L$ parameters, $\bar{u} \in [0, 1]^K$ and $\bar{v} \in [0, 1]^L$. The main challenge is to leverage this structure to learn efficiently. In this section, we discuss the challenges of solving our problem by existing algorithms. We conclude that a new algorithm is necessary and present it in Section 6.4.

Any rank-1 bandit is a multi-armed bandit with KL arms. As such, it can be solved by `UCB1` Auer et al. [2002a]. The n-step regret of `UCB1` in rank-1 bandits is $O(KL(1/\Delta)\log n)$. Therefore, `UCB1` is impractical when both K and L are large.

Note that $\log(\bar{u}(i)\bar{v}(j)) = \log(\bar{u}(i)) + \log(\bar{v}(j))$ for any $\bar{u}(i), \bar{v}(j) > 0$. Therefore, a rank-1 bandit can be viewed as a stochastic linear bandit and solved by `LinUCB` Dani et al. [2008], Abbasi-Yadkori et al. [2011], where the reward of arm (i, j) is $\log(\mathbf{u}_t(i)) + \log(\mathbf{v}_t(j))$ and its features $x_{i,j} \in \{0,1\}^{K+L}$ are

$$x_{i,j}(e) = \begin{cases} \mathbb{1}\{e = i\}, & e \leqslant K\,; \\ \mathbb{1}\{e - K = j\}, & e > K\,, \end{cases} \tag{6.1}$$

for any $e \in [K + L]$. This approach is problematic for at least two reasons. First, the reward is not properly defined when either $\mathbf{u}_t(i) = 0$ or $\mathbf{v}_t(j) = 0$. Second,

$$\mathbb{E}\left[\log(\mathbf{u}_t(i)) + \log(\mathbf{v}_t(j))\right] \neq \log(\bar{u}(i)) + \log(\bar{v}(j))\,.$$

Nevertheless, note that both sides of the above inequality have maxima at (i^*, j^*), and therefore `LinUCB` should perform well. We compare to it in Section 6.6.2.

Also note that $\bar{u}(i)\bar{v}(j) = \exp[\log(\bar{u}(i)) + \log(\bar{v}(j))]$ for $\bar{u}(i), \bar{v}(j) > 0$. Therefore, a rank-1 bandit can be viewed as a generalized linear bandit and solved by `GLM-UCB` Filippi et al. [2010], where the mean function is $\exp[\cdot]$ and the feature vector of arm (i, j) is in (6.1). This approach is not practical for three reasons. First, the parameter space is unbounded, because $\log(\bar{u}(i)) \to -\infty$ as $\bar{u}(i) \to 0$ and $\log(\bar{v}(j)) \to -\infty$ as $\bar{v}(j) \to 0$. Second, the confidence intervals of `GLM-UCB` are scaled by the reciprocal of the minimum derivative of the mean function c_μ^{-1}, which can be very large in our setting. In particular, $c_\mu = \min_{(i,j)\in[K]\times[L]} \bar{u}(i)\bar{v}(j)$. In addition, the gap-dependent upper bound on the regret of `GLM-UCB` is $O((K+L)^2 c_\mu^{-2})$, which further indicates that `GLM-UCB` is not practical. Our upper bound in Theorem 6.1 scales much better with all quantities of interest. Third, `GLM-UCB` needs to compute the maximum-likelihood estimates of $\bar{u}$ and $\bar{v}$ at each step, which is a non-convex optimization problem (Section 6.2).

Some variants of our problem can be solved trivially. For instance, let $\mathbf{u}_t(i) \in \{0.1, 0.5\}$ for all $i \in [K]$ and $\mathbf{v}_t(j) \in \{0.5, 0.9\}$ for all $j \in [L]$. Then $(\mathbf{u}_t(i), \mathbf{v}_t(j))$ can be identified from $\mathbf{u}_t(i)\mathbf{v}_t(j)$, and the learning problem does not seem more difficult than a stochastic combinatorial semi-bandit Kveton et al. [2015c]. We do not focus on such degenerate cases in this chapter.

6.4 `Rank1Elim` Algorithm

Our algorithm, `Rank1Elim`, is shown in Algorithm 6. It is an elimination algorithm Auer and Ortner [2010], which maintains `UCB1` confidence intervals Auer et al. [2002a] on the expected rewards of all rows and columns. `Rank1Elim` operates in stages, which quadruple in length. In each stage, it explores all remaining rows and columns randomly over all remaining columns and rows, respectively. At the end of the stage, it eliminates all rows and columns that cannot be optimal.

The eliminated rows and columns are tracked as follows. We denote by $\mathbf{h}^{\mathrm{u}}_{\ell}(i)$ the index of the most rewarding row whose expected reward is believed by `Rank1Elim` to be at least as high as that of row i in stage ℓ. Initially, $\mathbf{h}^{\mathrm{u}}_0(i) = i$. When row i is eliminated by row $\mathbf{i}_\ell$ in stage ℓ, $\mathbf{h}^{\mathrm{u}}_{\ell+1}(i)$ is set to $\mathbf{i}_\ell$; then when row $\mathbf{i}_\ell$ is eliminated by row $\mathbf{i}_{\ell'}$ in stage $\ell' > \ell$, $\mathbf{h}^{\mathrm{u}}_{\ell'+1}(i)$ is set to $\mathbf{i}_{\ell'}$; and so on. The corresponding column quantity, $\mathbf{h}^{\mathrm{v}}_{\ell}(j)$, is defined and updated analogously. The *remaining rows and columns in stage* ℓ, $\mathbf{I}_\ell$ and $\mathbf{J}_\ell$, are then the unique values in $\mathbf{h}^{\mathrm{u}}_{\ell}$ and $\mathbf{h}^{\mathrm{v}}_{\ell}$, respectively; and we set these in line 7 of Algorithm 6.

Each stage of Algorithm 6 has two main steps: exploration (lines 9–20) and elimination (lines 22–41). In the row exploration step, each row $i \in \mathbf{I}_\ell$ is explored randomly over all remaining columns $\mathbf{J}_\ell$ such that its expected reward up to stage ℓ is at least $\mu\bar{u}(i)$, where μ is in (6.4). To guarantee this, we sample column $j \in [L]$ randomly and then substitute it with column $\mathbf{h}^{\mathrm{v}}_{\ell}(j)$, which is at least as rewarding as column j. This is critical to avoid $1/\min_{j\in[L]}\bar{v}(j)$ in our regret bound, which can be large and is not necessary. The observations are stored in *reward matrix* $\mathbf{C}^{\mathrm{u}}_{\ell} \in \mathbb{R}^{K\times L}$. As all rows are explored similarly, their expected rewards are scaled similarly, and this permits elimination. The column exploration step is analogous.

In the elimination step, the confidence intervals of all remaining rows, $[\mathbf{L}^{\mathrm{u}}_\ell(i), \mathbf{U}^{\mathrm{u}}_\ell(i)]$ for any $i \in \mathbf{I}_\ell$, are estimated from matrix $\mathbf{C}^{\mathrm{u}}_\ell \in \mathbb{R}^{K \times L}$; and the confidence intervals of all remaining columns, $[\mathbf{L}^{\mathrm{v}}_\ell(j), \mathbf{U}^{\mathrm{v}}_\ell(j)]$ for any $j \in \mathbf{J}_\ell$, are estimated from $\mathbf{C}^{\mathrm{v}}_\ell \in \mathbb{R}^{K \times L}$. This separation is needed to guarantee that the expected rewards of all remaining rows and columns are scaled similarly. The confidence intervals are designed such that

$$\mathbf{U}^{\mathrm{u}}_\ell(i) \leqslant \mathbf{L}^{\mathrm{u}}_\ell(\mathbf{i}_\ell) = \max_{i \in \mathbf{I}_\ell} \mathbf{L}^{\mathrm{u}}_\ell(i)$$

implies that row i is suboptimal with a high probability for any column elimination policy up to the end of stage ℓ, and

$$\mathbf{U}^{\mathrm{v}}_\ell(j) \leqslant \mathbf{L}^{\mathrm{v}}_\ell(\mathbf{j}_\ell) = \max_{j \in \mathbf{J}_\ell} \mathbf{L}^{\mathrm{v}}_\ell(j)$$

implies that column j is suboptimal with a high probability for any row elimination policy up to the end of stage ℓ. As a result, all suboptimal rows and columns are eliminated correctly with a high probability.

6.5 Analysis

This section has three subsections. In Section 6.5.1, we derive a gap-dependent upper bound on the n-step regret of `Rank1Elim`. In Section 6.5.2, we derive a gap-dependent lower bound that nearly matches our upper bound. In Section 6.5.3, we discuss the results of our analysis.

6.5.1 Upper Bound

The hardness of our learning problem is measured by two sets of metrics. The first metrics are gaps. The *gaps* of row $i \in [K]$ and column $j \in [L]$ are defined as

$$\Delta^{\mathrm{u}}_i = \bar{u}(i^*) - \bar{u}(i)\,, \quad \Delta^{\mathrm{v}}_j = \bar{v}(j^*) - \bar{v}(j)\,, \tag{6.2}$$

respectively; and the *minimum row and column gaps* are defined as

$$\Delta^{\mathrm{u}}_{\min} = \min_{i \in [K] : \Delta^{\mathrm{u}}_i > 0} \Delta^{\mathrm{u}}_i \,, \quad \Delta^{\mathrm{v}}_{\min} = \min_{j \in [L] : \Delta^{\mathrm{v}}_j > 0} \Delta^{\mathrm{v}}_j \,, \tag{6.3}$$

respectively. Roughly speaking, the smaller the gaps, the harder the problem. The second metric is the minimum of the average of entries in $\bar{u}$ and $\bar{v}$, which is defined as

$$\mu = \min \left\{ \frac{1}{K} \sum_{i=1}^{K} \bar{u}(i), \, \frac{1}{L} \sum_{j=1}^{L} \bar{v}(j) \right\} . \tag{6.4}$$

The smaller the value of μ, the harder the problem. This quantity appears in our regret bound due to the averaging character of `Rank1Elim` (Section 6.4). Our upper bound on the regret of `Rank1Elim` is stated and proved below.

Theorem 6.1. *The expected* n*-step regret of* `Rank1Elim` *is bounded as*

$$R(n) \leqslant \frac{1}{\mu^2} \left(\sum_{i=1}^{K} \frac{384}{\bar{\Delta}^{\mathrm{u}}_i} + \sum_{j=1}^{L} \frac{384}{\bar{\Delta}^{\mathrm{v}}_j} \right) \log n + 3(K + L) \,,$$

where

$$\bar{\Delta}^{\mathrm{u}}_i = \Delta^{\mathrm{u}}_i + \mathbb{1}\{\Delta^{\mathrm{u}}_i = 0\} \, \Delta^{\mathrm{v}}_{\min} \,,$$
$$\bar{\Delta}^{\mathrm{v}}_j = \Delta^{\mathrm{v}}_j + \mathbb{1}\left\{\Delta^{\mathrm{v}}_j = 0\right\} \Delta^{\mathrm{u}}_{\min} \,.$$

The proof of Theorem 6.1 is organized as follows. First, we bound the probability that at least one confidence interval is violated. The corresponding regret is small, $O(K + L)$. Second, by the design of `Rank1Elim` and because all confidence intervals hold, the expected reward of any row $i \in [K]$ is at least $\mu \bar{u}(i)$. Because all rows are explored in the same way, any suboptimal row i is guaranteed to be eliminated after $O([1/(\mu \Delta^{\mathrm{u}}_i)^2] \log n)$ observations. Third, we factorize the regret due to exploring row i into its row and column components, and bound both of them. This is possible because `Rank1Elim` eliminates rows and columns simultaneously. Finally, we sum up the regret of all explored rows and columns.

Note that the gaps in Theorem 6.1, $\bar{\Delta}^{\mathrm{u}}_{i}$ and $\bar{\Delta}^{\mathrm{v}}_{j}$, are slightly different from those in (6.2). In particular, all zero row and column gaps in (6.2) are substituted with the minimum column and row gaps, respectively. The reason is that the regret due to exploring optimal rows and columns is positive until all suboptimal columns and rows are eliminated, respectively. The proof of Theorem 6.1 is below.

Proof. Let $\mathbf{R}^{\mathrm{u}}_{\ell}(i)$ and $\mathbf{R}^{\mathrm{v}}_{\ell}(j)$ be the stochastic regret associated with exploring row i and column j, respectively, in stage ℓ. Then the expected n-step regret of `Rank1Elim` is bounded as

$$R(n) \leqslant \mathbb{E}\left[\sum_{\ell=0}^{n-1}\left(\sum_{i=1}^{K}\mathbf{R}^{\mathrm{u}}_{\ell}(i) + \sum_{j=1}^{L}\mathbf{R}^{\mathrm{v}}_{\ell}(j)\right)\right]\,,$$

where the outer sum is over possibly n stages. Let

$$\begin{aligned}\bar{\mathbf{u}}_{\ell}(i) &= \sum_{t=0}^{\ell}\mathbb{E}\left[\sum_{j=1}^{L}\frac{\mathbf{C}^{\mathrm{u}}_{t}(i,j) - \mathbf{C}^{\mathrm{u}}_{t-1}(i,j)}{n_{\ell}}\,\middle|\,\mathbf{h}^{\mathrm{v}}_{t}\right] \\ &= \bar{u}(i)\sum_{t=0}^{\ell}\frac{n_t - n_{t-1}}{n_{\ell}}\sum_{j=1}^{L}\frac{\bar{v}(\mathbf{h}^{\mathrm{v}}_{t}(j))}{L}\end{aligned}$$

be the expected reward of row $i \in \mathbf{I}_{\ell}$ in the first ℓ stages, where $n_{-1} = 0$ and $\mathbf{C}^{\mathrm{u}}_{-1}(i,j) = 0$; and let

$$\mathcal{E}^{\mathrm{u}}_{\ell} = \{\forall i \in \mathbf{I}_{\ell} : \bar{\mathbf{u}}_{\ell}(i) \in [\mathbf{L}^{\mathrm{u}}_{\ell}(i), \mathbf{U}^{\mathrm{u}}_{\ell}(i)],\ \bar{\mathbf{u}}_{\ell}(i) \geqslant \mu\bar{u}(i)\}$$

be the event that for all remaining rows $i \in \mathbf{I}_{\ell}$ at the end of stage ℓ, the confidence interval on the expected reward holds and that this reward is at least $\mu\bar{u}(i)$. Let $\overline{\mathcal{E}^{\mathrm{u}}_{\ell}}$

be the complement of event $\mathcal{E}_\ell^u$. Let

$$\begin{aligned}\bar{\mathbf{v}}_\ell(j) &= \sum_{t=0}^{\ell} \mathbb{E}\left[\sum_{i=1}^{K} \frac{\mathbf{C}_t^v(i,j) - \mathbf{C}_{t-1}^v(i,j)}{n_\ell} \,\middle|\, \mathbf{h}_t^u\right] \\ &= \bar{v}(j) \sum_{t=0}^{\ell} \frac{n_t - n_{t-1}}{n_\ell} \sum_{i=1}^{K} \frac{\bar{u}(\mathbf{h}_t^u(i))}{K}\end{aligned}$$

denote the expected reward of column $j \in \mathbf{J}_\ell$ in the first ℓ stages, where $n_{-1} = 0$ and $\mathbf{C}_{-1}^v(i,j) = 0$; and let

$$\mathcal{E}_\ell^v = \{\forall j \in \mathbf{J}_\ell : \bar{\mathbf{v}}_\ell(j) \in [\mathbf{L}_\ell^v(j), \mathbf{U}_\ell^v(j)],\ \bar{\mathbf{v}}_\ell(j) \geqslant \mu\bar{v}(j)\}$$

be the event that for all remaining columns $j \in \mathbf{J}_\ell$ at the end of stage ℓ, the confidence interval on the expected reward holds and that this reward is at least $\mu\bar{v}(j)$. Let $\overline{\mathcal{E}_\ell^v}$ be the complement of event $\mathcal{E}_\ell^v$. Let $\mathcal{E}$ be the event that all events $\mathcal{E}_\ell^u$ and $\mathcal{E}_\ell^v$ happen; and $\overline{\mathcal{E}}$ be the complement of $\mathcal{E}$, the event that at least one of $\mathcal{E}_\ell^u$ and $\mathcal{E}_\ell^v$ does not happen. Then the expected n-step regret of `Rank1Elim` is bounded from above as

$$\begin{aligned}R(n) &\leqslant \mathbb{E}\left[\left(\sum_{\ell=0}^{n-1}\left(\sum_{i=1}^{K} \mathbf{R}_\ell^u(i) + \sum_{j=1}^{L} \mathbf{R}_\ell^v(j)\right)\right)\mathbb{1}\{\mathcal{E}\}\right] + \\ &\quad nP(\overline{\mathcal{E}}) \\ &\leqslant \sum_{i=1}^{K} \mathbb{E}\left[\sum_{\ell=0}^{n-1} \mathbf{R}_\ell^u(i)\mathbb{1}\{\mathcal{E}\}\right] + \\ &\quad \sum_{j=1}^{L} \mathbb{E}\left[\sum_{\ell=0}^{n-1} \mathbf{R}_\ell^v(j)\mathbb{1}\{\mathcal{E}\}\right] + 2(K+L)\,,\end{aligned}$$

where the last inequality is from the lemma proved in the Appendix.

Let $\mathcal{H}_\ell = (\mathbf{I}_\ell, \mathbf{J}_\ell)$ be the rows and columns in stage ℓ, and

$$\mathcal{F}_\ell = \left\{\forall i \in \mathbf{I}_\ell, j \in \mathbf{J}_\ell : \Delta_i^u \leqslant \frac{2\tilde{\Delta}_{\ell-1}}{\mu},\ \Delta_j^v \leqslant \frac{2\tilde{\Delta}_{\ell-1}}{\mu}\right\}$$

be the event that all rows and columns with "large gaps" are eliminated by the beginning of stage ℓ. By the lemma proved in the Appendix, event $\mathcal{E}$ causes event $\mathcal{F}_\ell$. Now note that the expected regret in stage ℓ is independent of $\mathcal{F}_\ell$ given $\mathcal{H}_\ell$. Therefore, the regret can be further bounded as

$$\begin{aligned} R(n) \leqslant{} & \sum_{i=1}^{K} \mathbb{E}\left[\sum_{\ell=0}^{n-1} \mathbb{E}\left[\mathbf{R}_\ell^{\mathrm{U}}(i) \,|\, \mathcal{H}_\ell\right] \mathbb{1}\{\mathcal{F}_\ell\}\right] + \\ & \sum_{j=1}^{L} \mathbb{E}\left[\sum_{\ell=0}^{n-1} \mathbb{E}\left[\mathbf{R}_\ell^{\mathrm{V}}(j) \,|\, \mathcal{H}_\ell\right] \mathbb{1}\{\mathcal{F}_\ell\}\right] + \\ & 2(K+L)\,. \end{aligned} \tag{6.5}$$

By the lemma proved in the Appendix,

$$\begin{aligned} \mathbb{E}\left[\sum_{\ell=0}^{n-1} \mathbb{E}\left[\mathbf{R}_\ell^{\mathrm{U}}(i) \,|\, \mathcal{H}_\ell\right] \mathbb{1}\{\mathcal{F}_\ell\}\right] &\leqslant \frac{384}{\mu^2 \bar{\Delta}_i^{\mathrm{U}}} \log n + 1\,, \\ \mathbb{E}\left[\sum_{\ell=0}^{n-1} \mathbb{E}\left[\mathbf{R}_\ell^{\mathrm{V}}(j) \,|\, \mathcal{H}_\ell\right] \mathbb{1}\{\mathcal{F}_\ell\}\right] &\leqslant \frac{384}{\mu^2 \bar{\Delta}_j^{\mathrm{V}}} \log n + 1\,, \end{aligned}$$

for any row $i \in [K]$ and column $j \in [L]$. Finally, we apply the above upper bounds to (6.5) and get our main claim. □

6.5.2 Lower Bound

We derive a gap-dependent lower bound on the family of rank-1 bandits where P_{U} and P_{V} are products of independent Bernoulli variables, which are parameterized by their means $\bar{u}$ and $\bar{v}$, respectively. The lower bound is derived for any *uniformly efficient algorithm $\mathcal{A}$*, which is any algorithm such that for any $(\bar{u}, \bar{v}) \in [0,1]^K \times [0,1]^L$ and any $\alpha \in (0,1)$, $R(n) = o(n^\alpha)$.

Theorem 6.2. *For any problem $(\bar{u}, \bar{v}) \in [0,1]^K \times [0,1]^L$ with a unique best arm and any* *uniformly efficient algorithm $\mathcal{A}$ whose regret is* $R(n)$,

$$\liminf_{n \to \infty} \frac{R(n)}{\log n} \geqslant \sum_{i \in [K] \setminus \{i^*\}} \frac{\bar{u}(i^*)\bar{v}(j^*) - \bar{u}(i)\bar{v}(j^*)}{d(\bar{u}(i)\bar{v}(j^*), \bar{u}(i^*)\bar{v}(j^*))} + \sum_{j \in [L] \setminus \{j^*\}} \frac{\bar{u}(i^*)\bar{v}(j^*) - \bar{u}(i^*)\bar{v}(j)}{d(\bar{u}(i^*)\bar{v}(j), \bar{u}(i^*)\bar{v}(j^*))},$$

where $d(p, q)$ *is the* Kullback-Leibler (KL) divergence *between Bernoulli random variables with means* p *and* q.

The lower bound involves two terms. The first term is the regret due to learning the optimal row i^*, while playing the optimal column j^*. The second term is the regret due to learning the optimal column j^*, while playing the optimal row i^*. We do not know whether this lower bound is tight. We discuss its tightness in Section 6.5.3.

Proof. The proof is based on the change-of-measure techniques from Kaufmann *et al.* Kaufmann et al. [2016] and Lagree *et al.* Lagree et al. [2016], who ultimately build on Graves and Lai Graves and Lai [1997]. Let

$$w^*(\bar{u}, \bar{v}) = \max_{(i,j) \in [K] \times [L]} \bar{u}(i)\bar{v}(j)$$

be the maximum reward in model $(\bar{u}, \bar{v})$. We consider the set of models where $\bar{u}(i^*)$ and $\bar{v}(j^*)$ remain the same, but the optimal arm changes,

$$B(\bar{u}, \bar{v}) = \{(\bar{u}', \bar{v}') \in [0,1]^K \times [0,1]^L : \bar{u}(i^*) = \bar{u}'(i^*), \bar{v}(j^*) = \bar{v}'(j^*),\ w^*(\bar{u}, \bar{v}) < w^*(\bar{u}', \bar{v}')\}.$$

By Theorem 17 of Kaufmann *et al.* Kaufmann et al. [2016],

$$\liminf_{n \to \infty} \frac{\sum_{i=1}^{K} \sum_{j=1}^{L} \mathbb{E}\left[\mathbf{T}_n(i,j)\right] d(\bar{u}(i)\bar{v}(j), \bar{u}'(i)\bar{v}'(j))}{\log n} \geqslant 1$$

for any $(\bar{u}', \bar{v}') \in B(\bar{u}, \bar{v})$, where $\mathbb{E}\left[\mathbf{T}_n(i,j)\right]$ is the expected number of times that arm (i,j) is chosen in n steps in problem $(\bar{u}, \bar{v})$. From this and the regret decomposition

$$R(n) = \textstyle\sum_{i=1}^{K} \sum_{j=1}^{L} \mathbb{E}\left[\mathbf{T}_n(i,j)\right] \left(\bar{u}(i^*)\bar{v}(j^*) - \bar{u}(i)\bar{v}(j)\right),$$

we get that

$$\liminf_{n \to \infty} \frac{R(n)}{\log n} \geqslant f(\bar{u}, \bar{v}),$$

where

$$\begin{aligned} f(\bar{u}, \bar{v}) = \inf_{c \in \Theta} \quad & \sum_{i=1}^{K} \sum_{j=1}^{L} (\bar{u}(i^*)\bar{v}(j^*) - \bar{u}(i)\bar{v}(j)) c_{i,j} \\ \text{s.t.} \quad & \forall (\bar{u}', \bar{v}') \in B(\bar{u}, \bar{v}) : \\ & \sum_{i=1}^{K} \sum_{j=1}^{L} d(\bar{u}(i)\bar{v}(j), \bar{u}'(i)\bar{v}'(j)) c_{i,j} \geqslant 1 \end{aligned}$$

and $\Theta = [0, \infty)^{K \times L}$. To obtain our lower bound, we carefully relax the constraints of the above problem, so that we do not loose much in the bound. The details are presented in the Appendix. In the relaxed problem, only $K + L - 1$ entries in the optimal solution c^* are non-zero, as in Combes *et al.* Combes et al. [2015a], and they are

$$c^*_{i,j} = \begin{cases} 1/d(\bar{u}(i)\bar{v}(j^*), \bar{u}(i^*)\bar{v}(j^*)), & j = j^*, i \neq i^*; \\ 1/d(\bar{u}(i^*)\bar{v}(j), \bar{u}(i^*)\bar{v}(j^*)), & i = i^*, j \neq j^*; \\ 0, & \text{otherwise.} \end{cases}$$

Now we substitute c^* into the objective of the above problem and get our lower bound. □

6.5.3 Discussion

We derive a gap-dependent upper bound on the n-step regret of `Rank1Elim` in Theorem 6.1, which is

$$O((K + L)(1/\mu^2)(1/\Delta) \log n)\,,$$

where K denotes the number of rows, L denotes the number of columns, $\Delta = \min\{\Delta^{\text{U}}_{\min}, \Delta^{\text{V}}_{\min}\}$ is the minimum of the row and column gaps in (6.3), and μ is the minimum of the average of entries in $\bar{u}$ and $\bar{v}$, as defined in (6.4).

We argue that our upper bound is nearly tight on the following class of problems. The i-th entry of $\mathbf{u}_t$, $\mathbf{u}_t(i)$, is an independent Bernoulli variable with mean

$$\bar{u}(i) = p_{\text{U}} + \Delta_{\text{U}} \mathbb{1}\{i = 1\}$$

for some $p_{\text{U}} \in [0, 1]$ and row gap $\Delta_{\text{U}} \in (0, 1 - p_{\text{U}}]$. The j-th entry of $\mathbf{v}_t$, $\mathbf{v}_t(j)$, is an independent Bernoulli variable with mean

$$\bar{v}(j) = p_{\text{V}} + \Delta_{\text{V}} \mathbb{1}\{j = 1\}$$

for $p_{\text{V}} \in [0, 1]$ and column gap $\Delta_{\text{V}} \in (0, 1 - p_{\text{V}}]$. Note that the optimal arm is $(1, 1)$ and that the expected reward for choosing it is $(p_{\text{U}} + \Delta_{\text{U}})(p_{\text{V}} + \Delta_{\text{V}})$. We refer to the instance of this problem by $B_{\text{SPIKE}}(K, L, p_{\text{U}}, p_{\text{V}}, \Delta_{\text{U}}, \Delta_{\text{V}})$; and parameterize it by K, L, p_{U}, p_{V}, Δ_{U}, and Δ_{V}.

Let $p_{\text{U}} = 0.5 - \Delta_{\text{U}}$ for $\Delta_{\text{U}} \in [0, 0.25]$, and $p_{\text{V}} = 0.5 - \Delta_{\text{V}}$ for $\Delta_{\text{V}} \in [0, 0.25]$. Then the upper bound in Theorem 6.1 is

$$O([K(1/\Delta_{\text{U}}) + L(1/\Delta_{\text{V}})] \log n)$$

since $1/\mu^2 \leqslant 1/0.25^2 = 16$. On the other hand, the lower bound in Theorem 6.2 is

$$\Omega([K(1/\Delta_{\text{U}}) + L(1/\Delta_{\text{V}})] \log n)$$

K	L	Regret
8	8	17491 ± 384
8	16	29628 ± 1499
8	32	50030 ± 1931
16	8	28862 ± 585
16	16	41823 ± 1689
16	32	62451 ± 2268
32	8	46156 ± 806
32	16	61992 ± 2339
32	32	85208 ± 3546

$p_U = p_V = 0.7, \ \Delta_U = \Delta_V = 0.2$

p_U	p_V	Regret
0.700	0.700	17744 ± 466
0.700	0.350	23983 ± 594
0.700	0.175	24776 ± 2333
0.350	0.700	22963 ± 205
0.350	0.350	38373 ± 71
0.350	0.175	57401 ± 68
0.175	0.700	27440 ± 2011
0.175	0.350	57492 ± 67
0.175	0.175	95586 ± 99

$\mathsf{K} = \mathsf{L} = 8, \ \Delta_U = \Delta_V = 0.2$

Δ_U	Δ_V	Regret
0.20	0.20	17653 ± 307
0.20	0.10	22891 ± 912
0.20	0.05	30954 ± 787
0.10	0.20	20958 ± 614
0.10	0.10	33642 ± 1089
0.10	0.05	45511 ± 3257
0.05	0.20	30688 ± 482
0.05	0.10	44390 ± 2542
0.05	0.05	68412 ± 2312

$\mathsf{K} = \mathsf{L} = 8, \ p_U = p_V = 0.7$

Table 6.1: The n-step regret of `Rank1Elim` in $n = 2\text{M}$ steps as K and L increase (left), p_U and p_V decrease (middle), and Δ_U and Δ_V decrease (right). The results are averaged over 20 runs.

since $d(p, q) \leqslant [q(1-q)]^{-1}(p-q)^2$ and $q = 1 - q = 0.5$. Note that the bounds match in K, L, the gaps, and $\log n$.

We conclude with the observation that `Rank1Elim` is suboptimal in problems where μ in (6.4) is small. In particular, consider the above problem, and choose $\Delta_U = \Delta_V = 0.5$ and $\mathsf{K} = \mathsf{L}$. In this problem, the regret of `Rank1Elim` is $O(\mathsf{K}^3 \log n)$; because `Rank1Elim` eliminates $O(\mathsf{K})$ rows and columns with $O(1/\mathsf{K})$ gaps, and the regret for choosing any suboptimal arm is $O(1)$. This is much higher than the regret of a naive solution by `UCB1` in Section 6.3, which would be $O(\mathsf{K}^2 \log n)$. Note that the upper bound in Theorem 6.1 is also $O(\mathsf{K}^3 \log n)$. Therefore, it is not loose, and a new algorithm is necessary to improve over `UCB1` in this particular problem.

6.6 Experiments

We conduct three experiments. In Section 6.6.1, we validate that the regret of `Rank1Elim` grows as suggested by Theorem 6.1. In Section 6.6.2, we compare `Rank1Elim` to three baselines. Finally, in Section 6.6.3, we evaluate `Rank1Elim` on a real-world problem where our modeling assumptions are violated.

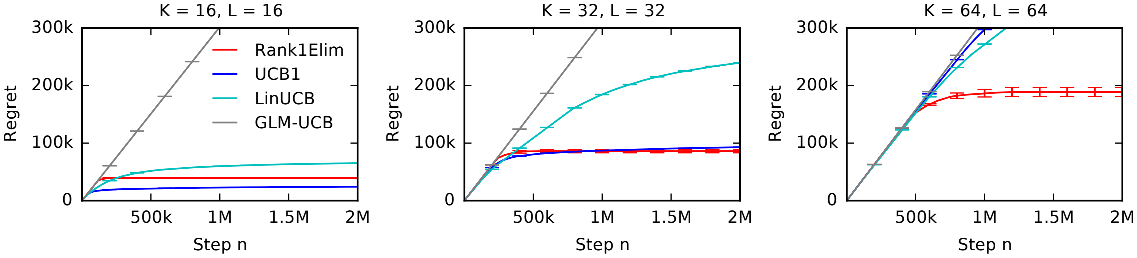


Figure 6.1: The n-step regret of `Rank1Elim`, `UCB1`, `LinUCB`, and `GLM-UCB` on three synthetic problems in up to $n = 2M$ steps. The results are averaged over 20 runs.

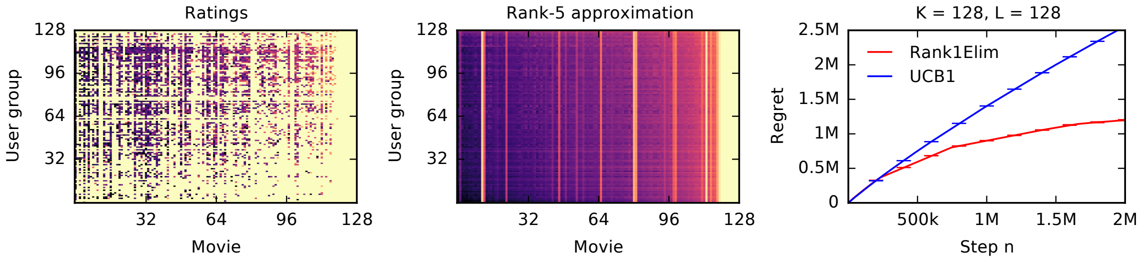
6.6.1 Regret Bound

The first experiment shows that the regret of `Rank1Elim` scales as suggested by our upper bound in Theorem 6.1. We experiment with the class of synthetic problems from Section 6.5.3, $B_{\text{SPIKE}}(K, L, p_U, p_V, \Delta_U, \Delta_V)$. We vary its parameters and report the n-step regret in 2 million (M) steps.

Table 6.1 shows the n-step regret of `Rank1Elim` for various choices of K, L, p_U, p_V, Δ_U, and Δ_V. In each table, we vary two parameters and keep the rest fixed. We observe that the regret increases as K and L increase, and Δ_U and Δ_V decrease; as suggested by Theorem 6.1. Specifically, the regret doubles when K and L are doubled, and when Δ_U and Δ_V are halved. We also observe that the regret is not quadratic in $1/\mu$, where $\mu \approx \min\{p_U, p_V\}$. This indicates that the upper bound in Theorem 6.1 is loose in μ when μ is bounded away from zero. We argue in Section 6.5.3 that this is not the case as $\mu \to 0$.

6.6.2 Comparison to Alternative Solutions

In the second experiment, we compare `Rank1Elim` to the three alternative methods in Section 6.3: `UCB1`, `LinUCB`, and `GLM-UCB`. The confidence radii of `LinUCB` and `GLM-UCB` are set as suggested by Abbasi-Yadkori *et al.* Abbasi-Yadkori et al. [2011] and Filippi *et al.* Filippi et al. [2010], respectively. The maximum-likelihood estimates of $\bar{u}$ and $\bar{v}$ in `GLM-UCB` are computed using the online EM Cappe and Moulines [2009], which is observed to converge to $\bar{u}$ and $\bar{v}$ in our problems. We experiment with the problem



(a)

(b)

(c)

Figure 6.2: **a**. Ratings from the MovieLens dataset. The darker the color, the higher the rating. The rows and columns are ordered by their average ratings. The missing ratings are shown in yellow. **b**. Rank-5 approximation to the ratings. **c**. The n-step regret of Rank1Elim and UCB1 in up to $n = 2$M steps.

from Section 6.6.1, where $p_U = p_V = 0.7$, $\Delta_U = \Delta_V = 0.2$, and $K = L$.

Our results are reported in Fig. 6.1. We observe that the regret of Rank1Elim flattens in all three problems, which indicates that Rank1Elim learns the optimal arm. When $K = 16$, UCB1 has a lower regret than Rank1Elim. However, because the regret of UCB1 is $O(KL)$ and the regret of Rank1Elim is $O(K + L)$, Rank1Elim can outperform UCB1 on larger problems. When $K = 32$, both algorithms already perform similarly; and when $K = 64$, Rank1Elim clearly outperforms UCB1. This shows that Rank1Elim can leverage the structure of our problem. Neither LinUCB nor GLM-UCB are competitive on any of our problems.

We investigated the poor performance of both LinUCB and GLM-UCB. When the confidence radii of LinUCB are multiplied by $1/3$, LinUCB becomes competitive on all problems. When the confidence radii of GLM-UCB are multiplied by $1/100$, GLM-UCB is still not competitive on any of our problems. We conclude that LinUCB and GLM-UCB perform poorly because their theory-suggested confidence intervals are too wide. In contrast, Rank1Elim is implemented with its theory-suggested intervals in all experiments.

6.6.3 MovieLens Experiment

In our last experiment, we evaluate Rank1Elim on a recommendation problem. The goal is to identify the pair of a user group and movie that has the highest expected rating. We experiment with the *MovieLens* dataset from February 2003 Lam and Herlocker [2013], where 6k users give 1M ratings to 4k movies.

Our learning problem is formulated as follows. We define a user group for every unique combination of gender, age group, and occupation in the MovieLens dataset. The total number of groups is 241. For each user group and movie, we average the ratings of all users in that group that rated that movie, and learn a low-rank approximation to the underlying rating matrix by a state-of-the-art algorithm Keshavan et al. [2010]. The algorithm automatically detects the rank of the matrix to be 5. We randomly choose $K = 128$ user groups and $L = 128$ movies. We report the average ratings of these user groups and movies in Fig. 6.2a, and the corresponding completed rating matrix in Fig. 6.2b. The reward for choosing user group $i \in [K]$ and movie $j \in [L]$ is a categorical random variable over five-star ratings. We estimate its parameters based on the assumption that the ratings are normally distributed with a fixed variance, conditioned on the completed ratings. The expected rewards in this experiment are not rank 1. Therefore, our model is misspecified and Rank1Elim has no guarantees on its performance.

Our results are reported in Fig. 6.2c. We observe that the regret of Rank1Elim is concave in the number of steps n, and flattens. This indicates that Rank1Elim learns a near-optimal solution. This is possible because of the structure of our rating matrix. Although it is rank 5, its first eigenvalue is an order of magnitude larger than the remaining four non-zero eigenvalues. This structure is not surprising because the ratings of items are often subject to significant *user and item biases* Koren et al. [2009]. Therefore, our rating matrix is nearly rank 1, and Rank1Elim learns a good solution. Our theory cannot explain this result and we leave it for future work. Finally, we note that UCB1 explores throughout because our problem has more than 10k arms.

6.7 Related Work

Zhao *et al.* Zhao et al. [2013] proposed a bandit algorithm for low-rank matrix completion, where the posterior of latent item factors is approximated by its point estimate. This algorithm is not analyzed. Kawale *et al.* Kawale et al. [2015] proposed a Thompson sampling (TS) algorithm for low-rank matrix completion, where the posterior of low-rank matrices is approximated by particle filtering. A computationally-inefficient variant of the algorithm has $O((1/\Delta^2) \log n)$ regret in rank-1 matrices. In contrast, note that `Rank1Elim` is computationally efficient and its n-step regret is $O((1/\Delta) \log n)$.

The problem of learning to recommended in the bandit setting was studied in several recent papers. Valko *et al.* Valko et al. [2014] and Kocak *et al.* Kocak et al. [2014] proposed content-based recommendation algorithms, where the features of items are derived from a known similarity graph over the items. Gentile *et al.* Gentile et al. [2014] proposed an algorithm that clusters users based on their preferences, under the assumption that the features of items are known. Li *et al.* Li et al. [2016a] extended this algorithm to the clustering of items. Maillard *et al.* Maillard and Mannor [2014] studied a multi-armed bandit problem where the arms are partitioned into latent groups. The problems in the last three papers are a special form of low-rank matrix completion, where some rows are identical. In this work, we do not make any such assumptions, but our results are limited to rank 1.

`Rank1Elim` is motivated by the structure of the position-based model Craswell et al. [2008]. Lagree *et al.* Lagree et al. [2016] proposed a bandit algorithm for this model under the assumption that the examination probabilities of all positions are known. Online learning to rank in click models was studied in several recent papers Kveton et al. [2015a], Combes et al. [2015a], Kveton et al. [2015b], Katariya et al. [2016], Li et al. [2016b], Zong et al. [2016]. In practice, the probability of clicking on an item depends on both the item and its position, and this work is a major step towards learning to rank from such heterogeneous effects.

6.8 Conclusions

In this work, we propose stochastic rank-1 bandits, a class of online learning problems where the goal is to learn the maximum entry of a rank-1 matrix. This problem is challenging because the reward is a product of latent random variables, which are not observed. We propose a practical algorithm for solving this problem, `Rank1Elim`, and prove a gap-dependent upper bound on its regret. We also prove a nearly matching gap-dependent lower bound. Finally, we evaluate `Rank1Elim` empirically. In particular, we validate the scaling of its regret, compare it to baselines, and show that it learns high-quality solutions even when our modeling assumptions are mildly violated.

We conclude that `Rank1Elim` is a practical algorithm for finding the maximum entry of a stochastic rank-1 matrix. It is surprisingly competitive with various baselines (Section 6.6.2) and can be applied to higher-rank matrices (Section 6.6.3). On the other hand, we show that `Rank1Elim` can be suboptimal on relatively simple problems (Section 6.5.3). We plan to address this issue in our future work. We note that our results can be generalized to other reward models, such as $\mathbf{u}_t(i)\mathbf{v}_t(j) \sim \mathcal{N}(\bar{u}(i)\bar{v}(j), \sigma)$ for $\sigma > 0$.

6.9 Appendix

6.9.1 Upper Bound

Lemma 6.3. *Let $\overline{\mathcal{E}}$ be defined as in the proof of Theorem 6.1. Then*

$$P(\overline{\mathcal{E}}) \leqslant \frac{2(K + L)}{n} \,.$$

Proof. Let $\mathcal{E}_\ell = \mathcal{E}_\ell^u \cap \mathcal{E}_\ell^v$. Then from the definition of $\overline{\mathcal{E}}$,

$$\overline{\mathcal{E}} = \overline{\mathcal{E}_0} \cup (\overline{\mathcal{E}_1} \cap \mathcal{E}_0) \cup \ldots \cup (\overline{\mathcal{E}_{n-1}} \cap \mathcal{E}_{n-2} \cap \ldots \cap \mathcal{E}_0) \,,$$

and from the definition of $\mathcal{E}_\ell$,

$$\overline{\mathcal{E}_\ell} \cap \mathcal{E}_{\ell-1} \cap \ldots \cap \mathcal{E}_0 = (\overline{\mathcal{E}_\ell^u} \cap \mathcal{E}_{\ell-1} \cap \ldots \cap \mathcal{E}_0) \cup (\overline{\mathcal{E}_\ell^v} \cap \mathcal{E}_{\ell-1} \cap \ldots \cap \mathcal{E}_0)\,.$$

It follows that the probability of event $\overline{\mathcal{E}}$ is bounded as

$$\begin{aligned} P(\overline{\mathcal{E}}) &\leqslant \sum_{\ell=0}^{n-1} P(\overline{\mathcal{E}_\ell^u},\, \mathcal{E}_0^u,\, \ldots,\, \mathcal{E}_{\ell-1}^u,\, \mathcal{E}_0^v,\, \ldots,\, \mathcal{E}_{\ell-1}^v) + \sum_{\ell=0}^{n-1} P(\overline{\mathcal{E}_\ell^v},\, \mathcal{E}_0^u,\, \ldots,\, \mathcal{E}_{\ell-1}^u,\, \mathcal{E}_0^v,\, \ldots,\, \mathcal{E}_{\ell-1}^v) \\ &\leqslant \sum_{\ell=0}^{n-1} P(\overline{\mathcal{E}_\ell^u},\, \mathcal{E}_0^v,\, \ldots,\, \mathcal{E}_{\ell-1}^v) + \sum_{\ell=0}^{n-1} P(\overline{\mathcal{E}_\ell^v},\, \mathcal{E}_0^u,\, \ldots,\, \mathcal{E}_{\ell-1}^u)\,. \end{aligned}$$

From the definition of $\overline{\mathcal{E}_\ell^u}$, it follows that

$$\begin{aligned} P(\overline{\mathcal{E}_\ell^u},\, \mathcal{E}_0^v,\, \ldots,\, \mathcal{E}_{\ell-1}^v) \leqslant{} & P(\exists i \in \mathbf{I}_\ell \text{ s.t. } \bar{\mathbf{u}}_\ell(i) \notin [\mathbf{L}_\ell^u(i), \mathbf{U}_\ell^u(i)]) + {} \\ & P(\exists i \in \mathbf{I}_\ell \text{ s.t. } \bar{\mathbf{u}}_\ell(i) < \mu \bar{u}(i),\, \mathcal{E}_0^v,\, \ldots,\, \mathcal{E}_{\ell-1}^v)\,. \end{aligned}$$

Now we bound the probability of the above two events. The probability $P(\overline{\mathcal{E}_\ell^v},\, \mathcal{E}_0^u,\, \ldots,\, \mathcal{E}_{\ell-1}^u)$ can be bounded similarly and we omit this proof.

Event 1: $\exists i \in \mathbf{I}_\ell$ s.t. $\bar{\mathbf{u}}_\ell(i) \notin [\mathbf{L}_\ell^u(i), \mathbf{U}_\ell^u(i)]$

Fix any $i \in \mathbf{I}_\ell$. Let $\mathbf{c}_k$ be the k-th observation of row i in the row exploration stage of `Rank1Elim` and $\ell(k)$ be the index of that stage. Then

$$\left(\mathbf{c}_k - \bar{u}(i) \sum_{j=1}^{L} \frac{\bar{v}(\mathbf{h}_{\ell(k)}^v(j))}{L}\right)_{k=1}^{n}$$

is a martingale difference sequence with respect to history $\mathbf{h}_0^v, \ldots, \mathbf{h}_{\ell(k)}^v$ in step k. This follows from the observation that

$$\mathbb{E}\left[\mathbf{c}_k \,\middle|\, \mathbf{h}_0^v, \ldots, \mathbf{h}_{\ell(k)}^v\right] = \bar{u}(i) \sum_{j=1}^{L} \frac{\bar{v}(\mathbf{h}_{\ell(k)}^v(j))}{L}\,,$$

because column $j \in [L]$ in stage $\ell(k)$ is chosen randomly and then mapped to at least as rewarding column $\mathbf{h}^{v}_{\ell(k)}(j)$. By the definition of our sequence and from the Azuma-Hoeffding inequality (Remark 2.2.1 of Raginsky and Sason Raginsky and Sason [2012]),

$$\begin{aligned}
P(\bar{\mathbf{u}}_\ell(i) \notin [\mathbf{L}^{u}_\ell(i), \mathbf{U}^{u}_\ell(i)]) &= P\left(\left|\frac{1}{n_\ell}\sum_{j=1}^{L} \mathbf{C}^{u}_\ell(i,j) - \bar{\mathbf{u}}_\ell(i)\right| > \sqrt{\frac{\log n}{n_\ell}}\right) \\
&= P\left(\left|\sum_{k=1}^{n_\ell}\left[\mathbf{c}_k - \bar{u}(i)\sum_{j=1}^{L}\frac{\bar{v}(\mathbf{h}^{v}_{\ell(k)}(j))}{L}\right]\right| > \sqrt{n_\ell \log n}\right) \\
&\leqslant 2\exp[-2\log n] \\
&= 2n^{-2}
\end{aligned}$$

for any stage ℓ. By the union bound,

$$P(\exists i \in \mathbf{I}_\ell \text{ s.t. } \bar{\mathbf{u}}_\ell(i) \notin [\mathbf{L}^{u}_\ell(i), \mathbf{U}^{u}_\ell(i)]) \leqslant 2Kn^{-2}$$

for any stage ℓ.

Event 2: $\exists i \in \mathbf{I}_\ell$ s.t. $\bar{\mathbf{u}}_\ell(i) < \mu\bar{u}(i)$, $\mathcal{E}^{v}_0$, $\ldots$, $\mathcal{E}^{v}_{\ell-1}$

We claim that this event cannot happen. Fix any $i \in \mathbf{I}_\ell$. When $\ell = 0$, we get that $\bar{\mathbf{u}}_0(i) = \bar{u}(i)(1/L)\sum_{j=1}^{L}\bar{v}(j) \geqslant \mu\bar{u}(i)$ from the definitions of $\bar{\mathbf{u}}_0(i)$ and μ, and event 2 obviously does not happen. When $\ell > 0$ and events $\mathcal{E}^{v}_0, \ldots, \mathcal{E}^{v}_{\ell-1}$ happen, any eliminated column j up to stage ℓ is substituted with column j' such that $\bar{v}(j') \geqslant \bar{v}(j)$, by the design of `Rank1Elim`. From this fact and the definition of $\bar{\mathbf{u}}_\ell(i)$, $\bar{\mathbf{u}}_\ell(i) \geqslant \mu\bar{u}(i)$. Therefore, event 2 does not happen when $\ell > 0$.

Total probability

Finally, we sum all probabilities up and get that

$$P(\overline{\mathcal{E}}) \leqslant n\left(\frac{2K}{n^2}\right) + n\left(\frac{2L}{n^2}\right) \leqslant \frac{2(K+L)}{n}.$$

This concludes our proof. □

Lemma 6.4. *Let event $\mathcal{E}$ happen and m be the first stage where $\tilde{\Delta}_m < \mu\Delta^{\mathrm{U}}_i/2$. Then row i is guaranteed to be eliminated by the end of stage m. Moreover, let m be the first stage where $\tilde{\Delta}_m < \mu\Delta^{\mathrm{V}}_j/2$. Then column j is guaranteed to be eliminated by the end of stage m.*

Proof. We only prove the first claim. The other claim is proved analogously.

Before we start, note that by the design of `Rank1Elim` and from the definition of m,

$$\tilde{\Delta}_m = 2^{-m} < \frac{\mu\Delta^{\mathrm{U}}_i}{2} \leqslant 2^{-(m-1)} = \tilde{\Delta}_{m-1}\,. \tag{6.6}$$

By the design of our confidence intervals,

$$\begin{aligned}
\frac{1}{n_m}\sum_{j=1}^{K}\mathbf{C}^{\mathrm{U}}_m(i,j) + \sqrt{\frac{\log n}{n_m}} &\overset{(a)}{\leqslant} \bar{\mathbf{u}}_m(i) + 2\sqrt{\frac{\log n}{n_m}} \\
&= \bar{\mathbf{u}}_m(i) + 4\sqrt{\frac{\log n}{n_m}} - 2\sqrt{\frac{\log n}{n_m}} \\
&\overset{(b)}{\leqslant} \bar{\mathbf{u}}_m(i) + 2\tilde{\Delta}_m - 2\sqrt{\frac{\log n}{n_m}} \\
&\overset{(c)}{\leqslant} \bar{\mathbf{u}}_m(i) + \mu\Delta^{\mathrm{U}}_i - 2\sqrt{\frac{\log n}{n_m}} \\
&= \bar{\mathbf{u}}_m(i^*) + \mu\Delta^{\mathrm{U}}_i - [\bar{\mathbf{u}}_m(i^*) - \bar{\mathbf{u}}_m(i)] - 2\sqrt{\frac{\log n}{n_m}}\,,
\end{aligned}$$

where inequality (a) is from $\mathbf{L}^{\mathrm{U}}_m(i) \leqslant \bar{\mathbf{u}}_m(i)$, inequality (b) is from $n_m \geqslant 4\tilde{\Delta}_m^{-2}\log n$, and inequality (c) is by (6.6). Now note that

$$\bar{\mathbf{u}}_m(i^*) - \bar{\mathbf{u}}_m(i) = q(\bar{u}(i^*) - \bar{u}(i)) \geqslant \mu\Delta^{\mathrm{U}}_i$$

for some $q \in [0,1]$. The equality holds because $\bar{\mathbf{u}}_m(i^*)$ and $\bar{\mathbf{u}}_m(i)$ are estimated from the same sets of random columns. The inequality follows from the fact that events $\mathcal{E}^{\mathrm{V}}_0, \dots, \mathcal{E}^{\mathrm{V}}_{m-1}$ happen. The events imply that any eliminated column j up to stage m is substituted with column j' such that $\bar{v}(j') \geqslant \bar{v}(j)$, and thus $q \geqslant \mu$. From the above inequality, we get that

$$\bar{\mathbf{u}}_m(i^*) + \mu\Delta^{\mathrm{U}}_i - [\bar{\mathbf{u}}_m(i^*) - \bar{\mathbf{u}}_m(i)] - 2\sqrt{\frac{\log n}{n_m}} \leqslant \bar{\mathbf{u}}_m(i^*) - 2\sqrt{\frac{\log n}{n_m}}\,.$$

Finally,

$$\begin{aligned}\bar{\mathbf{u}}_m(i^*) - 2\sqrt{\frac{\log n}{n_m}} &\overset{(a)}{\leqslant} \frac{1}{n_m}\sum_{j=1}^{K} \mathbf{C}^{\mathrm{U}}_m(i^*, j) - \sqrt{\frac{\log n}{n_m}} \\ &\overset{(b)}{\leqslant} \frac{1}{n_m}\sum_{j=1}^{K} \mathbf{C}^{\mathrm{U}}_m(\mathbf{i}_m, j) - \sqrt{\frac{\log n}{n_m}}\,,\end{aligned}$$

where inequality (a) follows from $\bar{\mathbf{u}}_m(i^*) \leqslant \mathbf{U}^{\mathrm{U}}_m(i^*)$ and inequality (b) follows from $\mathbf{L}^{\mathrm{U}}_m(i^*) \leqslant \mathbf{L}^{\mathrm{U}}_m(\mathbf{i}_m)$, since $i^* \in \mathbf{I}_m$ and $\mathbf{i}_m = \arg\max_{i \in \mathbf{I}_m} \mathbf{L}^{\mathrm{U}}_m(i)$. Now we chain all inequalities and get our final claim. □

Lemma 6.5. *The expected cumulative regret due to exploring any row* $i \in [K]$ *and any column* $j \in [L]$ *is bounded as*

$$\begin{aligned}\mathbb{E}\left[\sum_{\ell=0}^{n-1} \mathbb{E}\left[\mathbf{R}^{\mathrm{U}}_\ell(i) \mid \mathcal{H}_\ell\right] \mathbb{1}\{\mathcal{F}_\ell\}\right] &\leqslant \frac{384}{\mu^2 \bar{\Delta}^{\mathrm{U}}_i} \log n + 1\,, \\ \mathbb{E}\left[\sum_{\ell=0}^{n-1} \mathbb{E}\left[\mathbf{R}^{\mathrm{V}}_\ell(j) \mid \mathcal{H}_\ell\right] \mathbb{1}\{\mathcal{F}_\ell\}\right] &\leqslant \frac{384}{\mu^2 \bar{\Delta}^{\mathrm{V}}_j} \log n + 1\,.\end{aligned}$$

Proof. We only prove the first claim. The other claim is proved analogously. This proof has two parts. In the first part, we assume that row i is suboptimal, $\Delta^{\mathrm{U}}_i > 0$. In the second part, we assume that row i is optimal, $\Delta^{\mathrm{U}}_i = 0$.

Row i is suboptimal

Let row i be suboptimal and m be the first stage where $\tilde{\Delta}_m < \mu\Delta^{\mathrm{U}}_i/2$. Then row

i is guaranteed to be eliminated by the end of stage m (Lemma 6.4), and thus

$$\mathbb{E}\left[\sum_{\ell=0}^{n-1}\mathbb{E}\left[\mathbf{R}_\ell^u(i)\,|\,\mathcal{H}_\ell\right]\mathbb{1}\{\mathcal{F}_\ell\}\right] \leqslant \mathbb{E}\left[\sum_{\ell=0}^{m}\mathbb{E}\left[\mathbf{R}_\ell^u(i)\,|\,\mathcal{H}_\ell\right]\mathbb{1}\{\mathcal{F}_\ell\}\right] .$$

By Lemma 6.6, the expected regret of choosing row i in stage ℓ can be bounded from above as

$$\mathbb{E}\left[\mathbf{R}_\ell^u(i)\,|\,\mathcal{H}_\ell\right]\mathbb{1}\{\mathcal{F}_\ell\} \leqslant (\Delta_i^u + \max_{j\in \mathbf{J}_\ell}\Delta_j^v)(n_\ell - n_{\ell-1})\,,$$

where $\max_{j\in \mathbf{J}_\ell}\Delta_j^v$ is the maximum column gap in stage ℓ, n_ℓ is the number of steps by the end of stage ℓ, and $n_{-1} = 0$. From the definition of $\mathcal{F}_\ell$ and $\tilde{\Delta}_\ell$, if column j is not eliminated before stage ℓ, we have that

$$\Delta_j^v \leqslant \frac{2\tilde{\Delta}_{\ell-1}}{\mu} = \frac{2\cdot 2^{m-\ell+1}\tilde{\Delta}_m}{\mu} < 2^{m-\ell+1}\Delta_i^u\,.$$

From the above inequalities and the definition of n_ℓ, it follows that

$$\begin{aligned}
\mathbb{E}\left[\sum_{\ell=0}^{m}\mathbb{E}\left[\mathbf{R}_\ell^u(i)\,|\,\mathcal{H}_\ell\right]\mathbb{1}\{\mathcal{F}_\ell\}\right] &\leqslant \sum_{\ell=0}^{m}(\Delta_i^u + \max_{j\in \mathbf{J}_\ell}\Delta_j^v)(n_\ell - n_{\ell-1}) \\
&\leqslant \sum_{\ell=0}^{m}(\Delta_i^u + 2^{m-\ell+1}\Delta_i^u)(n_\ell - n_{\ell-1}) \\
&\leqslant \Delta_i^u\left(n_m + \sum_{\ell=0}^{m}2^{m-\ell+1}n_\ell\right) \\
&\leqslant \Delta_i^u\left(2^{2m+2}\log n + 1 + \sum_{\ell=0}^{m}2^{m-\ell+1}(2^{2\ell+2}\log n + 1)\right) \\
&= \Delta_i^u\left(2^{2m+2}\log n + 1 + \sum_{\ell=0}^{m}2^{m+\ell+3}\log n + \sum_{\ell=0}^{m}2^{m-\ell+1}\right) \\
&\leqslant \Delta_i^u(5\cdot 2^{2m+2}\log n + 2^{m+2}) + 1 \\
&\leqslant 6\cdot 2^4\cdot 2^{2m-2}\Delta_i^u\log n + 1\,,
\end{aligned}$$

where the last inequality follows from $\log n \geqslant 1$ for $n \geqslant 3$. From the definition of $\tilde{\Delta}_{m-1}$ in (6.6), we have that

$$2^{m-1} = \frac{1}{\tilde{\Delta}_{m-1}} \leqslant \frac{2}{\mu \Delta^{u}_{i}} .$$

Now we chain all above inequalities and get that

$$\mathbb{E}\left[\sum_{\ell=0}^{n-1} \mathbb{E}\left[\mathbf{R}^{u}_{\ell}(i) \,|\, \mathcal{H}_{\ell}\right] \mathbb{1}\{\mathcal{F}_{\ell}\}\right] \leqslant 6 \cdot 2^{4} \cdot 2^{2m-2} \Delta^{u}_{i} \log n + 1 \leqslant \frac{384}{\mu^{2} \Delta^{u}_{i}} \log n + 1 .$$

This concludes the first part of our proof.

Row i is optimal

Let row i be optimal and m be the first stage where $\tilde{\Delta}_{m} < \mu \Delta^{v}_{\min}/2$. Then similarly to the first part of the analysis,

$$\mathbb{E}\left[\sum_{\ell=0}^{n-1} \mathbb{E}\left[\mathbf{R}^{u}_{\ell}(i) \,|\, \mathcal{H}_{\ell}\right] \mathbb{1}\{\mathcal{F}_{\ell}\}\right] \leqslant \sum_{\ell=0}^{m} (\max_{j \in \mathbf{J}_{\ell}} \Delta^{v}_{j})(n_{\ell} - n_{\ell-1}) \leqslant \frac{384}{\mu^{2} \Delta^{v}_{\min}} \log n + 1 .$$

This concludes our proof. □

Lemma 6.6. *Let* $\mathbf{u} \sim P_{u}$ *and* $\mathbf{v} \sim P_{v}$ *be drawn independently. Then the expected regret of choosing any row* $i \in [K]$ *and column* $j \in [L]$ *is bounded from above as*

$$\mathbb{E}\left[\mathbf{u}(i^{*})\mathbf{v}(j^{*}) - \mathbf{u}(i)\mathbf{v}(j)\right] \leqslant \Delta^{u}_{i} + \Delta^{v}_{j} .$$

Proof. Note that for any $x, y, x^{*}, y^{*} \in [0, 1]$,

$$x^{*}y^{*} - xy = x^{*}y^{*} - xy^{*} + xy^{*} - xy = y^{*}(x^{*} - x) + x(y^{*} - y) \leqslant (x^{*} - x) + (y^{*} - y) .$$

By the independence of the entries of $\mathbf{u}$ and $\mathbf{v}$, and from the above inequality,

$$\mathbb{E}\left[\mathbf{u}(i^{*})\mathbf{v}(j^{*}) - \mathbf{u}(i)\mathbf{v}(j)\right] = \bar{u}(i^{*})\bar{v}(j^{*}) - \bar{u}(i)\bar{v}(j) \leqslant (\bar{u}(i^{*}) - \bar{u}(i)) + (\bar{v}(j^{*}) - \bar{v}(j)) .$$

This concludes our proof. □

6.9.2 Lower Bound

In this section we present the missing details of the proof of Theorem 6.2. Recall that we need to bound from below the value of $f(\bar{u},\bar{v})$ where

$$\begin{aligned} f(\bar{u},\bar{v}) = \inf_{c\in[0,\infty)^{K\times L}} & \sum_{i=1}^{K}\sum_{j=1}^{L}(\bar{u}(i^*)\bar{v}(j^*) - \bar{u}(i)\bar{v}(j))c_{i,j} \\ \text{s.t. } & \forall(\bar{u}',\bar{v}') \in B(\bar{u},\bar{v}) : \\ & \quad \sum_{i=1}^{K}\sum_{j=1}^{L} d(\bar{u}(i)\bar{v}(j), \bar{u}'(i)\bar{v}'(j))c_{i,j} \geqslant 1 \end{aligned}$$

and

$$B(\bar{u},\bar{v}) = \{(\bar{u}',\bar{v}') \in [0,1]^K \times [0,1]^L : \bar{u}(i^*) = \bar{u}'(i^*), \, \bar{v}(j^*) = \bar{v}'(j^*), \, w^*(\bar{u},\bar{v}) < w^*(\bar{u}',\bar{v}')\}.$$

Without loss of generality, we assume that the optimal action in the original model $(\bar{u},\bar{v})$ is $(i^*,j^*) = (1,1)$. Moreover, we consider a class of *identifiable* bandit models, meaning that we assume that

$$\forall(i,i',j,j') \in [0,1]^{2K} \times [0,1]^{2L}, \quad (i,j) \neq (i',j') \implies 0 < d(\bar{u}(i)\bar{v}(j), \bar{u}(i')\bar{v}(j')) < +\infty.$$

This implies in particular that $\bar{u}(i^*)\bar{v}(j^*)$ must be less than 1. An intuitive justification of this assumption is the following. Remark that for the Bernoulli problem we consider here, if the mean of the best arm is exactly 1, the rewards from optimal pulls are always 1 so that the empirical average is always exactly 1 and as we cap the UCBs to 1, the optimal arm is always a candidate to the next pull, which leads to constant regret. Also note that by our assumption, the optimal action is unique. To get a lower bound, we consider the same optimization problem as above, but replace B with its subset. Clearly, this can only decrease the optimal value.

Concretely, we consider only those models in $B(\bar{u},\bar{v})$ where only one parameter changes at a time. Let

$$\begin{aligned}
B_u(\bar{u},\bar{v}) &= \{(\bar{u}',\bar{v}) : \bar{u}' \in [0,1]^K,\ \exists i_0 \in \{2,\dots,K\}, \epsilon \in [0,1] \text{ s.t. } [\forall i \neq i_0 : \bar{u}'(i) = \bar{u}(i)] \text{ and } \bar{u}'(i_0) = \bar{u}(1)\\
B_v(\bar{u},\bar{v}) &= \{(\bar{u},\bar{v}') : \bar{v}' \in [0,1]^L,\ \exists j_0 \in \{2,\dots,L\}, \epsilon \in [0,1] \text{ s.t. } [\forall j \neq j_0 : \bar{v}'(j) = \bar{v}(j)] \text{ and } \bar{v}'(j_0) = \bar{v}(1) +
\end{aligned}$$

Let $f'(\bar{u},\bar{v})$ be the optimal value of the above optimization problem when $B(\bar{u},\bar{v})$ is replaced by $B_u(\bar{u},\bar{v}) \cup B_v(\bar{u},\bar{v}) \subset B(\bar{u},\bar{v})$. Now suppose that $(\bar{u}',\bar{v}') \in B_u(\bar{u},\bar{v})$ and $i_0 = 2$. Then, for any $i \neq 2$ and $j \in [L]$, $d(\bar{u}(i)\bar{v}(j), \bar{u}'(i)\bar{v}'(j)) = 0$; and for $i = 2$ and any $j \in [L]$, $d(\bar{u}(i)\bar{v}(j), \bar{u}'(i)\bar{v}'(j)) = d(\bar{u}(2)\bar{v}(j), (\bar{u}(1)+\epsilon)\bar{v}(j))$. Hence,

$$\sum_{i=1}^{K}\sum_{j=1}^{L} d(\bar{u}(i)\bar{v}(j), \bar{u}'(i)\bar{v}'(j)) = \sum_{j=1}^{L} d(\bar{u}(2)\bar{v}(j), (\bar{u}(1)+\epsilon)\bar{v}(j))\,.$$

Reasoning similarly for $B_v(\bar{u},\bar{v})$, we see that $f'(\bar{u},\bar{v})$ satisfies

$$\begin{aligned}
f'(\bar{u},\bar{v}) = \inf_{c \in [0,\infty)^{K\times L}} & \sum_{i=1}^{K}\sum_{j=1}^{L} (\bar{u}(i^*)\bar{v}(j^*) - \bar{u}(i)\bar{v}(j))c_{i,j}\\
\text{s.t.} \quad & \forall \epsilon_v \in (0, 1-\bar{v}(1)], \epsilon_u \in (0, 1-\bar{u}(1)]\\
& \quad \forall j \neq 1, \sum_{i=1}^{K} d(\bar{u}(i)\bar{v}(j), \bar{u}(i)(\bar{v}(1)+\epsilon_v))c_{i,j} \geqslant 1\\
& \quad \forall i \neq 1, \sum_{j=1}^{L} d(\bar{u}(i)\bar{v}(j), (\bar{u}(1)+\epsilon_u)\bar{v}(j))c_{i,j} \geqslant 1.
\end{aligned}$$

Clearly, the smaller the coefficients of $c_{i,j}$ in the constraints, the tighter the constraints. We obtain the smallest coefficients when $\epsilon_v, \epsilon_u \to 0$. By continuity, we get

$$\begin{aligned} f'(\bar{u},\bar{v}) = \inf_{c\in[0,\infty)^{K\times L}} & \sum_{i=1}^{K}\sum_{j=1}^{L}(\bar{u}(i^*)\bar{v}(j^*) - \bar{u}(i)\bar{v}(j))c_{i,j} \\ \text{s.t.} \quad & \forall j \neq 1, \sum_{i=1}^{K} d(\bar{u}(i)\bar{v}(j), \bar{u}(i)\bar{v}(1))c_{i,j} \geqslant 1 \\ & \forall i \neq 1, \sum_{j=1}^{L} d(\bar{u}(i)\bar{v}(j), \bar{u}(1)\bar{v}(j))c_{i,j} \geqslant 1. \end{aligned}$$

Let

$$c_{i,j} = \begin{cases} 1/d(\bar{u}(i)\bar{v}(1), \bar{u}(1)\bar{v}(1))\,, & j = 1 \text{ and } i > 1\,; \\ 1/d(\bar{u}(1)\bar{v}(j), \bar{u}(1)\bar{v}(1))\,, & i = 1 \text{ and } j > 1\,; \\ 0\,, & \text{otherwise.} \end{cases}$$

We claim that $(c_{i,j})$ is an optimal solution for the problem defining f'.

First, we show that $(c_{i,j})$ is feasible. Let $i \neq 1$. Then $\sum_{j=1}^{L} d(\bar{u}(i)\bar{v}(j), \bar{u}(1)\bar{v}(j))c_{i,j} = d(\bar{u}(i)\bar{v}(1), \bar{u}(1)\bar{v}(1))c_{i,1} = 1$. Similarly, we can verify the other constraint, too, showing that $(c_{i,j})$ is indeed feasible.

Now, it remains to show that the proposed solution is indeed optimal. We prove this by contradiction, following the ideas of Combes et al. [2015a]. We suppose that there exists a solution c of the optimization problem such that $c_{i_0,j_0} > 0$ for $i_0 \neq 1$ and $j_0 \neq 1$. Then, we prove that it is possible to find another feasible solution c' but with an objective lower than that obtained with c, contradicting the assumption of optimality of c.

We define c' as follows, redistributing the mass of c_{i_0,j_0} on the first row and the first column:

$$c'_{i,j} = \begin{cases} 0\,, & i = i_0 \text{ and } j = j_0\,; \\ c_{i_0,1} + c_{i_0,j_0} \dfrac{d(\bar{u}(i_0)\bar{v}(j_0), \bar{u}(1)\bar{v}(j_0))}{d(\bar{u}(i_0)\bar{v}(1), \bar{u}(1)\bar{v}(1))}\,, & i = i_0 \text{ and } j = 1\,; \\ \\ c_{1,j_0} + c_{i_0,j_0} \dfrac{d(\bar{u}(i_0)\bar{v}(j_0), \bar{u}(i_0)\bar{v}(1))}{d(\bar{u}(1)\bar{v}(j_0), \bar{u}(1)\bar{v}(1))}\,, & i = 1 \text{ and } j = j_0\,; \\ c_{i,j}\,, & \text{otherwise.} \end{cases}$$

It is easily verified that if c satisfies the constraints, then so does c' because the missing mass of c_{i_0,j_0} is simply redistributed on $c'_{i_0,1}$ and c'_{1,j_0}. For example, for $i = i_0$ we have

$$\begin{aligned} &\sum_{j=1}^{L} d(\bar{u}(i_0)\bar{v}(j), \bar{u}(1)\bar{v}(j))c'_{i_0,j} - \sum_{j=1}^{L} d(\bar{u}(i_0)\bar{v}(j), \bar{u}(1)\bar{v}(j))c_{i_0,j} \\ &= d(\bar{u}(i_0)\bar{v}(1), \bar{u}(1)\bar{v}(1))c_{i_0,j_0} \frac{d(\bar{u}(i_0)\bar{v}(j_0), \bar{u}(1)\bar{v}(j_0))}{d(\bar{u}(i_0)\bar{v}(1), \bar{u}(1)\bar{v}(1))} - c_{i_0,j_0} d(\bar{u}(i_0)\bar{v}(j_0), \bar{u}(1)\bar{v}(j_0)) \\ &= 0 \end{aligned}$$

while for $i \notin \{1, i_0\}$, $c'_{i,j} = c_{i,j}$, so $\sum_{j=1}^{L} d(\bar{u}(i)\bar{v}(j), \bar{u}(1)\bar{v}(j))c'_{i,j} = \sum_{j=1}^{L} d(\bar{u}(i)\bar{v}(j), \bar{u}(1)\bar{v}(j))c_{i,j}$.

Now, we prove that the objective function is lower for c' than for c by showing that the difference between them is negative:

$$\begin{aligned}
\Delta &\doteq \sum_{i=1}^{K}\sum_{j=1}^{L}(\bar{u}(1)\bar{v}(1)-\bar{u}(i)\bar{v}(j))c'_{i,j} - \sum_{i=1}^{K}\sum_{j=1}^{L}(\bar{u}(1)\bar{v}(1)-\bar{u}(i)\bar{v}(j))c_{i,j} \\
&= \quad c_{i_0,j_0}(\bar{u}(1)\bar{v}(1)-\bar{u}(i_0)\bar{v}(1))\frac{d(\bar{u}(i_0)\bar{v}(j_0),\bar{u}(1)\bar{v}(j_0))}{d(\bar{u}(i_0)\bar{v}(1),\bar{u}(1)\bar{v}(1))} \\
&\quad + c_{i_0,j_0}(\bar{u}(1)\bar{v}(1)-\bar{u}(1)\bar{v}(j_0))\frac{d(\bar{u}(i_0)\bar{v}(j_0),\bar{u}(i_0)\bar{v}(1))}{d(\bar{u}(1)\bar{v}(j_0),\bar{u}(1)\bar{v}(1))} \\
&\quad - c_{i_0,j_0}(\bar{u}(1)\bar{v}(1)-\bar{u}(i_0)\bar{v}(j_0)) \\
&= \quad c_{i_0,j_0}\Big\{(\bar{u}(1)-\bar{u}(i_0))\bar{v}(1)\frac{d(\bar{u}(i_0)\bar{v}(j_0),\bar{u}(1)\bar{v}(j_0))}{d(\bar{u}(i_0)\bar{v}(1),\bar{u}(1)\bar{v}(1))} \\
&\qquad + (\bar{v}(1)-\bar{v}(j_0))\bar{u}(1)\frac{d(\bar{u}(i_0)\bar{v}(j_0),\bar{u}(i_0)\bar{v}(1))}{d(\bar{u}(1)\bar{v}(j_0),\bar{u}(1)\bar{v}(1))} \\
&\qquad - (\bar{u}(1)\bar{v}(1)-\bar{u}(i_0)\bar{v}(j_0))\Big\}
\end{aligned}$$

Writing

$$\bar{u}(1)\bar{v}(1)-\bar{u}(i_0)\bar{v}(j_0) = (\bar{u}(1)-\bar{u}(i_0))\bar{v}(j_0) + (\bar{v}(1)-\bar{v}(j_0))\bar{u}(1)$$

we get

$$\begin{aligned}
\Delta = c_{i_0,j_0}(\bar{u}(1)-\bar{u}(i_0))&\left(\bar{v}(1)\frac{d(\bar{u}(i_0)\bar{v}(j_0),\bar{u}(1)\bar{v}(j_0))}{d(\bar{u}(i_0)\bar{v}(1),\bar{u}(1)\bar{v}(1))} - \bar{v}(j_0)\right) \\
+ c_{i_0,j_0}(\bar{v}(1)-\bar{v}(j_0))&\left(\bar{u}(1)\frac{d(\bar{u}(i_0)\bar{v}(j_0),\bar{u}(i_0)\bar{v}(1))}{d(\bar{u}(1)\bar{v}(j_0),\bar{u}(1)\bar{v}(1))} - \bar{u}(1)\right).
\end{aligned}$$

To finish the proof, it suffices to prove that both terms of the above sum are negative. First, $\bar{u}(1)-\bar{u}(i_0), \bar{v}(1)-\bar{v}(j_0), c_{i_0,j_0} > 0$, hence it remains to consider the terms involving the ratios of KL divergences. Note that both ratios take the form $\frac{d(\alpha p,\alpha q)}{d(p,q)}$ with $\alpha < 1$, but one must be compared to $\alpha < 1$ while the other can simply be compared to 1. For the first such term, showing the negativity of the difference is equivalent to showing that for $\alpha = \bar{v}(j_0)/\bar{v}(1) < 1$,

$$\frac{d(\alpha\bar{u}(i_0)\bar{v}(1), \alpha\bar{u}(1)\bar{v}(1))}{d(\bar{u}(i_0)\bar{v}(1), \bar{u}(1)\bar{v}(1))} < \alpha.$$

Lemma 6.7 below shows that for fixed $(p, q) \in (0, 1)^2$, $f : \alpha \mapsto d(\alpha p, \alpha q)$ is convex, which proves the above inequality. For the second term, it remains to see whether the ratio of the KL divergences is below one. Lemma 6.7 proven below shows that the function $\alpha \mapsto d(\alpha p, \alpha q)$ is increasing on $(0, 1)$, showing that

$$\frac{d(\bar{u}(i_0)\bar{v}(j_0), \bar{u}(i_0)\bar{v}(1))}{d(\bar{u}(1)\bar{v}(j_0), \bar{u}(1)\bar{v}(1))} < 1\,.$$

Thus, the proof is finished once we prove Lemma 6.7.

Lemma 6.7. *Let* p, q *be any fixed real numbers in* $(0, 1)$. *The function* $f : \alpha \mapsto d(\alpha p, \alpha q)$ *is convex and increasing on* $(0, 1)$. *As a consequence, for any* $\alpha < 1$, $d(\alpha p, \alpha q) < d(p, q)$.

Proof. We first re-parametrize our problem into polar coordinates (r, θ) :

$$\begin{cases} p &= r\cos\theta \\ q &= r\sin\theta \end{cases}$$

In order to prove the statement of the lemma, it now suffices to prove that $f_\theta : r \mapsto d(r\sin\theta, r\cos\theta)$ is increasing. We have

$$f_\theta(r) = r\cos\theta\log\left(\frac{\cos\theta}{\sin\theta}\right) + (1 - r\cos\theta)\log\left(\frac{1 - r\cos\theta}{1 - r\sin\theta}\right)$$

which can be differentiated along r for a fixed θ :

$$f'_\theta(r) = \cos\theta\log\left(\frac{1 - r\sin\theta}{1 - r\cos\theta}\right) + \frac{\sin\theta - \cos\theta}{1 - r\sin\theta} + \cos\theta\log\left(\frac{\cos\theta}{\sin\theta}\right).$$

Now, we can differentiate again along r and after some calculations we obtain

$$f''_\theta(r) = \frac{(\sin\theta - \cos\theta)^2}{(1 - r\sin\theta)^2(1 - r\cos\theta)} > 0$$

which proves that the function f_θ is convex. It remains to prove that $f'_\theta(0) \geqslant 0$ for any $\theta \in (0, \pi/2)$. We rewrite $f'_\theta(0)$ as a function of θ :

$$\begin{aligned} f'_\theta(0) &= \cos\theta \log\left(\frac{\cos\theta}{\sin\theta}\right) + \sin\theta - \cos\theta \\ &:= \phi(\theta) \end{aligned}$$

Let us assume that there exists $\theta_0 \in (0, \pi/2)$ such that $\phi(\theta_0) < 0$. Then, in this direction $f'_\theta(0) < 0$ and as $f_\theta(0) = 0$ for any $\theta \in (0, \pi/2)$, it means that there exists $r_0 > 0$ such that $f_{\theta_0}(r_0) < 0$. Yet, $f_{\theta_0}(r_0) = d(r_0\cos\theta_0, r_0\sin\theta_0) > 0$ because of the positivity of the KL divergence.

So by contradiction, we proved that for all $\theta \in (0, \pi/2)$, $f'_\theta(0) = \phi(\theta) \geqslant 0$ and by convexity f_θ is non-negative and non-decreasing on $[0, +\infty)$.

□

6.9.3 Gaussian payoffs

The lower bound naturally extends to other classes of distributions, such as Gaussians. For illustration here we show the lower bound for this case. We still assume that the means are in $[0, 1]$, as before. We also assume that all payoffs have a common variance $\sigma^2 > 0$. Recall that the Kullback-Leibler divergence between two distributions with fixed variance σ^2 is $d(p, q) = (p - q)^2/(2\sigma^2)$. Then, the proof of Theorem 6.2 can be repeated with minor differences (in particular, the proof of the analogue of Lemma 6.7 becomes trivial) and we get the following result:

Theorem 6.8. *For any $(\bar{u}, \bar{v}) \in [0, 1]^K \times [0, 1]^L$ with a unique optimal action and any uniformly efficient algorithm $\mathcal{A}$ whose regret is $R(n)$, assuming Gaussian row and column* *rewards with common variance* σ^2*,*

$$\liminf_{n\to\infty} \frac{R(n)}{\log(n)} \geqslant \frac{2\sigma^2}{\bar{v}(j^*)} \sum_{i\in[K]\setminus\{i^*\}} \frac{1}{\Delta_i^u} + \frac{2\sigma^2}{\bar{u}(i^*)} \sum_{j\in[L]\setminus\{j^*\}} \frac{1}{\Delta_j^v} \,.$$

Algorithm 6 `Rank1Elim` for stochastic rank-1 bandits.

```
// Initialization
$t \leftarrow 1,\ \tilde{\Delta}_0 \leftarrow 1,\ \mathbf{C}^u_0 \leftarrow \{0\}^{K \times L},\ \mathbf{C}^v_0 \leftarrow \{0\}^{K \times L},$
$\mathbf{h}^u_0 \leftarrow (1, \dots, K),\ \mathbf{h}^v_0 \leftarrow (1, \dots, L),\ n_{-1} \leftarrow 0$

for all $\ell = 0, 1, \dots$ do
    $n_\ell \leftarrow \lceil 4 \tilde{\Delta}_\ell^{-2} \log n \rceil$
    $\mathbf{I}_\ell \leftarrow \bigcup_{i \in [K]} \{\mathbf{h}^u_\ell(i)\},\ \mathbf{J}_\ell \leftarrow \bigcup_{j \in [L]} \{\mathbf{h}^v_\ell(j)\}$

    // Row and column exploration
    for $n_\ell - n_{\ell-1}$ times do
        Choose uniformly at random column $j \in [L]$
        $j \leftarrow \mathbf{h}^v_\ell(j)$
        for all $i \in \mathbf{I}_\ell$ do
            $\mathbf{C}^u_\ell(i, j) \leftarrow \mathbf{C}^u_\ell(i, j) + \mathbf{u}_t(i) \mathbf{v}_t(j)$
            $t \leftarrow t + 1$
        end for
        Choose uniformly at random row $i \in [K]$
        $i \leftarrow \mathbf{h}^u_\ell(i)$
        for all $j \in \mathbf{J}_\ell$ do
            $\mathbf{C}^v_\ell(i, j) \leftarrow \mathbf{C}^v_\ell(i, j) + \mathbf{u}_t(i) \mathbf{v}_t(j)$
            $t \leftarrow t + 1$
        end for
    end for

    // UCBs and LCBs on the expected rewards of all remaining rows and columns
    for all $i \in \mathbf{I}_\ell$ do
        $\mathbf{U}^u_\ell(i) \leftarrow \frac{1}{n_\ell} \sum_{j=1}^{L} \mathbf{C}^u_\ell(i, j) + \sqrt{\frac{\log n}{n_\ell}}$
        $\mathbf{L}^u_\ell(i) \leftarrow \frac{1}{n_\ell} \sum_{j=1}^{L} \mathbf{C}^u_\ell(i, j) - \sqrt{\frac{\log n}{n_\ell}}$
    end for
    for all $j \in \mathbf{J}_\ell$ do
        $\mathbf{U}^v_\ell(j) \leftarrow \frac{1}{n_\ell} \sum_{i=1}^{K} \mathbf{C}^v_\ell(i, j) + \sqrt{\frac{\log n}{n_\ell}}$
        $\mathbf{L}^v_\ell(j) \leftarrow \frac{1}{n_\ell} \sum_{i=1}^{K} \mathbf{C}^v_\ell(i, j) - \sqrt{\frac{\log n}{n_\ell}}$
    end for

    // Row and column elimination
    $\mathbf{i}_\ell \leftarrow \arg\max_{i \in \mathbf{I}_\ell} \mathbf{L}^u_\ell(i)$
    $\mathbf{h}^u_{\ell+1} \leftarrow \mathbf{h}^u_\ell$
    for all $i = 1, \dots, K$ do
        if $\mathbf{U}^u_\ell(\mathbf{h}^u_\ell(i)) \leqslant \mathbf{L}^u_\ell(\mathbf{i}_\ell)$ then
            $\mathbf{h}^u_{\ell+1}(i) \leftarrow \mathbf{i}_\ell$
        end if
    end for

    $\mathbf{j}_\ell \leftarrow \arg\max_{j \in \mathbf{J}_\ell} \mathbf{L}^v_\ell(j)$
    $\mathbf{h}^v_{\ell+1} \leftarrow \mathbf{h}^v_\ell$
    for all $j = 1, \dots, L$ do
        if $\mathbf{U}^v_\ell(\mathbf{h}^v_\ell(j)) \leqslant \mathbf{L}^v_\ell(\mathbf{j}_\ell)$ then
            $\mathbf{h}^v_{\ell+1}(j) \leftarrow \mathbf{j}_\ell$
        end if
    end for

    $\tilde{\Delta}_{\ell+1} \leftarrow \tilde{\Delta}_\ell / 2,\ \mathbf{C}^u_{\ell+1} \leftarrow \mathbf{C}^u_\ell,\ \mathbf{C}^v_{\ell+1} \leftarrow \mathbf{C}^v_\ell$
end for
```

7 BERNOULLI RANK-1 BANDITS FOR CLICK FEEDBACK

7.1 Introduction

When deciding which search results to present, click logs are of particular interest. A fundamental problem in click data is position bias. The probability of an element being clicked depends not only on its relevance, but also on its position on the results page. The position-based model (PBM), first proposed by Richardson *et al.* Richardson et al. [2007] and then formalized by Craswell *et al.* Craswell et al. [2008], models this behavior by associating with each item a probability of being *attractive*, and with each position a probability of being *examined*. To be clicked, a result must be both attractive and examined. Given click logs, the attraction and examination probabilities can be learned using the maximum-likelihood estimation (MLE) or the expectation-maximization (EM) algorithms Chuklin et al. [2015b].

An online learning model for this problem is proposed in Katariya *et al.* Katariya et al. [2017], called *stochastic rank*-1 *bandit*. The objective of the learning agent is to learn the most rewarding item and position, which is the maximum entry of a rank-1 matrix. At time t, the agent chooses a pair of row and column arms, and receives the product of their values as a reward. The goal of the agent is to maximize its expected cumulative reward, or equivalently to minimize its expected cumulative regret with respect to the optimal solution, the most rewarding pair of row and column arms. This learning problem is challenging because when the agent receives the reward of zero, it could mean either that the item was unattractive, or the position was not examined, or both.

Katariya *et al.* Katariya et al. [2017] also proposed an elimination algorithm, `Rank1Elim`, whose regret is $\mathcal{O}((K+L)\,\mu^{-2}\Delta^{-1}\log n)$, where K is the number of rows, L is the number of columns, Δ is the minimum of the row and column gaps, and μ is the minimum of the average row and column rewards. When μ is bounded away from zero, the regret scales linearly with $K+L$, while it scales inversely with Δ. This is a significant improvement over using a standard bandit algorithm that (disregarding the problem structure) would treat item-position pairs as unrelated arms and would achieve a regret of $O(KL\Delta^{-1}\log n)$. The issue is that as μ gets small, the regret bound worsens significantly. As we verify in Section 7.5, this indeed happens on models derived from some real-world problems. To illustrate the severity of this problem, consider as an example where $K = L$, and the row and column rewards are Bernoulli distributed. Let the mean reward of row 1 and column 1 be Δ, and the mean reward of all other rows and columns be 0. We refer to this setting as a "needle in a haystack", because there is a single rewarding entry out of K^2 entries. For this setting, $\mu = \Delta/K$, and consequently the regret of `Rank1Elim` is $\mathcal{O}(\mu^{-2}\Delta^{-1}K\log n) = \mathcal{O}(K^3\log n)$. However, a naive bandit algorithm that ignores the rank-1 structure and treats each row-column pair as unrelated arms has $\mathcal{O}(K^2\log n)$ regret.[1] While a naive bandit algorithm is unable to exploit the rank-1 structure when μ is large, `Rank1Elim` is unable to keep up with a naive algorithm when μ is small. Our goal in this chapter is to design an algorithm that performs well across all rank-1 problem instances regardless of their parameters.

In this chapter we propose that this improvement can be achieved by replacing the "`UCB1` confidence intervals" used by `Rank1Elim` by strictly tighter confidence intervals based on Kullback-Leibler (KL) divergences. This leads to our algorithm that we call `Rank1ElimKL`. Based on the work of Garivier and Cappe Garivier and Cappe [2011], we expect this change to lead to an improved behavior, especially for extreme instances, as $\mu \to 0$. Indeed, in this chapter we show that KL divergences enjoy a peculiar "scaling". In particular, thanks to this improvement, for the "needle in a haystack" problem discussed above the regret of `Rank1ElimKL` becomes $\mathcal{O}(K^2\log n)$.

Our contributions are as follows. First, we propose a *Bernoulli rank-*1 *bandit*, which is a special class of a *stochastic rank-*1 *bandit* where the rewards are Bernoulli distributed. Second, we modify `Rank1Elim` for solving the Bernoulli rank-1 bandit, which we call `Rank1ElimKL`, to use `KL-UCB` intervals. Third, we derive a $O((K + L)\,(\mu\gamma\Delta)^{-1}\log n)$ gap-dependent upper bound on the n-step regret of `Rank1ElimKL`, where K, L, Δ and μ are as above, while $\gamma = \max\{\mu, 1 - p_{\max}\}$ with $p_{\max}$ being the

[1] Alternatively, the worst-case regret bound for `Rank1Elim` becomes $O(Kn^{2/3}\log n)$, while that of for a naive bandit algorithm with a naive bound is $O(Kn^{1/2}\log n)$.

maximum of the row and column rewards; effectively replacing the μ^{-2} term of the previous regret bound of `Rank1Elim` with $(\mu\gamma)^{-1}$. It follows that the new bound is an unilateral improvement over the previous one and is a strict improvement when $\mu < 1 - p_{\max}$, which is expected to happen quite often in practical problems. For the "needle in a haystack" problem, the new bound essentially matches that of the naive bandit algorithm, while never worsening the bound of `Rank1Elim`. Our final contribution is the experimental validation of `Rank1ElimKL`, on both synthetic and real-world problems. The experiments indicate that `Rank1ElimKL` outperforms several baselines across almost all problem instances.

We denote random variables by boldface letters and define $[n] = \{1, \dots, n\}$. For any sets A and B, we denote by A^B the set of all vectors whose entries are indexed by B and take values from A. We let $d(p, q) = p \log \frac{p}{q} + (1 - p) \log \frac{1-p}{1-q}$ denote the KL divergence between the Bernoulli distributions with means $p, q \in [0, 1]$. As usual, the formula for $d(p, q)$ is defined through its continuous extension as p, q approach the boundaries of $[0, 1]$.

7.2 Setting

The setting of the *Bernoulli rank-*1 *bandit* is the same as that of the stochastic rank-1 bandit Katariya et al. [2017], with the additional requirement that the row and column rewards are Bernoulli distributed. We state the setting for completeness, and borrow the notation from Katariya *et al.* Katariya et al. [2017] for the ease of comparison.

An instance of our learning problem is defined by a tuple (K, L, P_U, P_V), where K is the number of rows, L is the number of columns, P_U is a distribution over $\{0, 1\}^K$ from which the row rewards are drawn, and P_V is a distribution over $\{0, 1\}^L$ from which the column rewards are drawn.

Let the row and column rewards be

$$(\mathbf{u}_t, \mathbf{v}_t) \overset{\text{i.i.d}}{\sim} P_U \otimes P_V \,, \qquad t = 1, \dots, n \,.$$

In particular, $\mathbf{u}_t$ and $\mathbf{v}_t$ are drawn independently at any time t. At time t, the learning agent chooses a row index $\mathbf{i}_t \in [K]$ and a column index $\mathbf{j}_t \in [L]$, and observes $\mathbf{u}_t(\mathbf{i}_t)\mathbf{v}(\mathbf{j}_t)$ as its reward. The indices $\mathbf{i}_t$ and $\mathbf{j}_t$ chosen by the learning agent are allowed to depend only on the history of the agent up to time t.

Let the time horizon be n. The goal of the agent is to maximize its expected cumulative reward in n steps. This is equivalent to minimizing the *expected cumulative regret* in n steps

$$R(n) = \mathbb{E}\left[\sum_{t=1}^{n} R(\mathbf{i}_t, \mathbf{j}_t, \mathbf{u}_t, \mathbf{v}_t)\right],$$

where $R(\mathbf{i}_t, \mathbf{j}_t, \mathbf{u}_t, \mathbf{v}_t) = \mathbf{u}_t(i^*)\mathbf{v}_t(j^*) - \mathbf{u}_t(\mathbf{i}_t)\mathbf{v}_t(\mathbf{j}_t)$ is the *instantaneous stochastic regret* of the agent at time t, and

$$(i^*, j^*) = \arg\max_{(i,j)\in[K]\times[L]} \mathbb{E}\left[\mathbf{u}(i)\mathbf{v}(j)\right]$$

is the *optimal solution* in hindsight of knowing P_U and P_V.

7.3 `Rank1ElimKL` Algorithm

The pseudocode of our algorithm, `Rank1ElimKL`, is in Algorithm 7. As noted earlier this algorithm is based on `Rank1Elim` Katariya et al. [2017] with the difference that we replace their confidence intervals with KL-based confidence intervals. For the reader's benefit, we explain the full algorithm.

`Rank1ElimKL` is an elimination algorithm that operates in stages, where the elimination is conducted with `KL-UCB` confidence intervals. The lengths of the stages quadruple from one stage to the next, and the algorithm is designed such that at the end of stage ℓ, it eliminates with high probability any row and column whose gap scaled by a problem dependent constant is at least $\tilde{\Delta}_\ell = 2^{-\ell}$. We denote the *remaining rows and columns* in stage ℓ by $\mathbf{I}_\ell$ and $\mathbf{J}_\ell$, respectively.

Every stage has an exploration phase and an exploitation phase. During row-exploration in stage ℓ (lines 12–16), every remaining row is played with a randomly chosen remaining column, and the rewards are added to the table $\mathbf{C}^{u}_{\ell} \in \mathbb{R}^{K \times L}$. Similarly, during column-exploration in stage ℓ (lines 17–21), every remaining column is played with a randomly chosen remaining row, and the rewards are added to the table $\mathbf{C}^{v}_{\ell} \in \mathbb{R}^{K \times L}$. We play every row (column) with the same random column (row), and separate the row and column reward tables, so that the expected rewards of any two rows (columns) are scaled by the same quantity at the end of any phase. This facilitates comparison between rows (columns) and elimination in the exploitation phase. The distributions used in selecting random columns and rows are such that the row (column) means do not decrease over time.

In the exploitation phase, we construct high-probability KL-UCB Garivier and Cappe [2011] confidence intervals $[\mathbf{L}^{u}_{\ell}(i), \mathbf{U}^{u}_{\ell}(i)]$ for row $i \in \mathbf{I}_{\ell}$, and confidence intervals $[\mathbf{L}^{v}_{\ell}(j), \mathbf{U}^{v}_{\ell}(j)]$ for column $j \in \mathbf{J}_{\ell}$. As noted earlier, this is where we depart from Rank1Elim. The elimination uses row $\mathbf{i}_{\ell}$ and column $\mathbf{j}_{\ell}$, where

$$\mathbf{i}_{\ell} = \arg\max_{i \in \mathbf{I}_{\ell}} \mathbf{L}^{u}_{\ell}(i)\,, \qquad \mathbf{j}_{\ell} = \arg\max_{j \in \mathbf{J}_{\ell}} \mathbf{L}^{v}_{\ell}(j)\,.$$

We eliminate any row i and column j such that

$$\mathbf{U}^{u}_{\ell}(i) \leqslant \mathbf{L}^{u}_{\ell}(\mathbf{i}_{\ell})\,, \qquad \mathbf{U}^{v}_{\ell}(j) \leqslant \mathbf{L}^{v}_{\ell}(\mathbf{j}_{\ell})\,.$$

We also track the remaining rows and columns in stage ℓ by $\mathbf{h}^{u}_{\ell}$ and $\mathbf{h}^{v}_{\ell}$, respectively. When row i is eliminated by row $\mathbf{i}_{\ell}$, we set $\mathbf{h}^{u}_{\ell}(i) = \mathbf{i}_{\ell}$. If row $\mathbf{i}_{\ell}$ is eliminated by row $\mathbf{i}_{\ell'}$ at a later stage $\ell' > \ell$, we update $\mathbf{h}^{u}_{\ell}(i) = \mathbf{i}_{\ell'}$. This is analogous for columns. The remaining rows $\mathbf{I}_{\ell}$ and columns $\mathbf{J}_{\ell}$ can be then defined as the unique values in $\mathbf{h}^{u}_{\ell}$ and $\mathbf{h}^{v}_{\ell}$, respectively. The maps $\mathbf{h}^{u}_{\ell}$ and $\mathbf{h}^{v}_{\ell}$ help to guarantee that the row and column means are non-decreasing.

The KL-UCB confidence intervals in Rank1ElimKL can be found by solving a one-dimensional convex optimization problem for every row (lines 27–28) and column (lines 31–32). They can be found efficiently using binary search because the Kullback-Leibler divergence $d(x, q)$ is convex in q as q moves away from x in either direction.

The KL-UCB confidence intervals need to be computed only once per stage. Hence, Rank1ElimKL has to solve at most $K + L$ convex optimization problems per stage, and hence $(K + L) \log n$ problems overall.

7.4 Analysis

In this section, we derive a gap-dependent upper bound on the n-step regret of Rank1ElimKL. The hardness of our learning problem is measured by two kinds of metrics. The first kind are gaps. The *gaps* of row $i \in [K]$ and column $j \in [L]$ are defined as

$$\Delta_i^u = \bar{u}(i^*) - \bar{u}(i)\,, \quad \Delta_j^v = \bar{v}(j^*) - \bar{v}(j)\,, \tag{7.1}$$

respectively; and the *minimum row and column gaps* are defined as

$$\Delta_{\min}^u = \min_{i \in [K] : \Delta_i^u > 0} \Delta_i^u\,, \quad \Delta_{\min}^v = \min_{j \in [L] : \Delta_j^v > 0} \Delta_j^v\,, \tag{7.2}$$

respectively. Roughly speaking, the smaller the gaps, the harder the problem. This inverse dependence on gaps is tight Katariya et al. [2017].

The second kind of metrics are the extremal parameters

$$\mu = \min\left\{\frac{1}{K}\sum_{i=1}^{K} \bar{u}(i),\ \frac{1}{L}\sum_{j=1}^{L} \bar{v}(j)\right\}\,, \tag{7.3}$$

$$p_{\max} = \max\left\{\max_{i \in [K]} \bar{u}(i),\ \max_{j \in [L]} \bar{v}(j)\right\}\,. \tag{7.4}$$

The first metric, μ, is the minimum of the average of entries of $\bar{u}$ and $\bar{v}$. This quantity appears in our analysis due to the averaging character of Rank1ElimKL. The smaller the value of μ, the larger the regret. The second metric, $p_{\max}$, is the maximum entry in $\bar{u}$ and $\bar{v}$. As we shall see the regret scales inversely with

$$\gamma = \max\{\mu, 1 - p_{\max}\}\,. \tag{7.5}$$

Note that if $\mu \to 0$ and $p_{\max} \to 1$ at the same time, then the row and columns gaps must also approach one. With this we are ready to state our main result.

Theorem 7.1. *Let* $C = 6e{+}82$ *and* $n \geqslant 5$. *Then the expected* n*-step regret of* `Rank1ElimKL` *is bounded as*

$$R(n) \leqslant \frac{160}{\mu\gamma}\left(\sum_{i=1}^{K} \frac{1}{\bar{\Delta}_i^u} + \sum_{j=1}^{L} \frac{1}{\bar{\Delta}_j^v}\right) \log n + C(K + L)\,,$$

where

$$\bar{\Delta}_i^u = \Delta_i^u + \mathbb{1}\{\Delta_i^u = 0\}\,\Delta_{\min}^v\,,$$
$$\bar{\Delta}_j^v = \Delta_j^v + \mathbb{1}\left\{\Delta_j^v = 0\right\}\,\Delta_{\min}^u\,.$$

The difference from the main result of Katariya *et al.* Katariya et al. [2017] is that the first term in our bound scales with $1/(\mu\gamma)$ instead of $1/\mu^2$. Since $\mu \leqslant \gamma$ and in fact often $\mu \ll \gamma$, this is a significant improvement. We validate this empirically in the next section.

Due to the lack of space, we only provide a sketch of the proof of Theorem 7.1. At a high level, it follows the steps of the proof of Katariya *et al.* Katariya et al. [2017]. Focusing on the source of the improvement, we first state and prove a new lemma, which allows us to replace one $1/\mu$ in the regret bound with $1/\gamma$. Recall from Section 7.1 that d denotes the KL divergence between Bernoulli random variables with means $p, q \in [0, 1]$.

Lemma 7.2. *Let* $c, p, q \in [0, 1]$. *Then*

$$c(1 - \max\{p, q\})d(p, q) \leqslant d(cp, cq) \leqslant c d(p, q)\,. \tag{7.6}$$

In particular,

$$2c \max(c, 1 - \max\{p, q\})(p - q)^2 \leqslant d(cp, cq)\,. \tag{7.7}$$

Proof. The proof of (7.6) is based on differentiation. The first two derivatives of $d(cp, cq)$ with respect to q are

$$\frac{\partial}{\partial q} d(cp, cq) = \frac{c(q-p)}{q(1-cq)},$$
$$\frac{\partial^2}{\partial q^2} d(cp, cq) = \frac{c^2(q-p)^2 + cp(1-cp)}{q^2(1-cq)^2};$$

and the first two derivatives of $cd(p, q)$ with respect to q are

$$\frac{\partial}{\partial q}[cd(p, q)] = \frac{c(q-p)}{q(1-q)},$$
$$\frac{\partial^2}{\partial q^2}[cd(p, q)] = \frac{c(q-p)^2 + cp(1-p)}{q^2(1-q)^2}.$$

The second derivatives show that both $d(cp, cq)$ and $cd(p, q)$ are convex in q for any p. The minima are at $q = p$.

We fix p and c, and prove (7.6) for any q. The upper bound is derived as follows. Since

$$d(cp, cx) = cd(p, x) = 0$$

when $x = p$, the upper bound holds if $cd(p, x)$ increases faster than $d(cp, cx)$ for any $p < x \leqslant q$, and if $cd(p, x)$ decreases faster than $d(cp, cx)$ for any $q \leqslant x < p$. This follows from the definitions of $\frac{\partial}{\partial x} d(cp, cx)$ and $\frac{\partial}{\partial x}[cd(p, x)]$. In particular, both derivatives have the same sign for any x, and $1/(1-cx) \leqslant 1/(1-x)$ for $x \in [\min\{p, q\}, \max\{p, q\}]$.

The lower bound is derived as follows. Note that the ratio of $\frac{\partial}{\partial x}[cd(p, x)]$ and $\frac{\partial}{\partial x} d(cp, cx)$ is bounded from above as

$$\frac{\frac{\partial}{\partial x}[cd(p, x)]}{\frac{\partial}{\partial x} d(cp, cx)} = \frac{1-cx}{1-x} \leqslant \frac{1}{1-x} \leqslant \frac{1}{1-\max\{p, q\}}$$

for any $x \in [\min\{p, q\}, \max\{p, q\}]$. Therefore, we get a lower bound on $d(cp, cq)$

when we multiply $cd(p, q)$ by $1 - \max\{p, q\}$.

To prove (7.7) note that by Pinsker's inequality, for any p, q, $d(p, q) \geqslant 2(p - q)^2$. Hence, on one hand, $d(cp, cq) \geqslant 2c^2(p-q)^2$. On the other hand, we have from (7.6) that $d(cp, cq) \geqslant 2c(1-\max\{p, q\})(p-q)^2$. Taking the maximum of the right-hand sides in these two equations gives (7.7). □

Proof sketch of Theorem 7.1. We proceed along the lines of Katariya *et al.* Katariya et al. [2017]. The key step in their analysis is the upper bound on the expected n-step regret of any suboptimal row $i \in [K]$. This bound is proved as follows. First, Katariya *et al.* Katariya et al. [2017] show that row i is eliminated with a high probability after $O((\mu\Delta_i^U)^{-2} \log n)$ observations, for any column elimination strategy. Then they argue that the amortized per-observation regret before the elimination is $O(\Delta_i^U)$. Therefore, the maximum regret due to row i is $O(\mu^{-2}(\Delta_i^U)^{-1} \log n)$. The expected n-step regret of any suboptimal column $j \in [L]$ is bounded analogously.

We modify the above argument as follows. Roughly speaking, due to the `KL-UCB` confidence interval, a suboptimal row i is eliminated with a high probability after

$$O\left(\frac{1}{d(\mu(\bar{u}(i^*) - \Delta_i^U), \mu\bar{u}(i^*))} \log n\right)$$

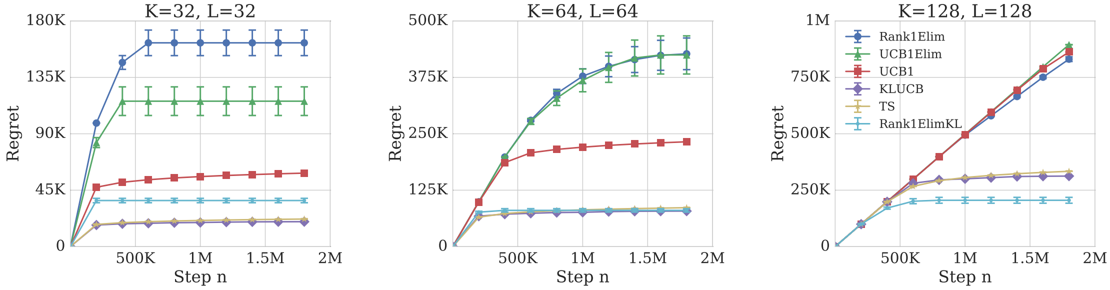
observations. Therefore, the expected n-step regret due to exploring row i is

$$O\left(\frac{\Delta_i^U}{d(\mu(\bar{u}(i^*) - \Delta_i^U), \mu\bar{u}(i^*))} \log n\right) .$$

Now we apply (7.7) of Lemma 7.2 to get that the regret is

$$O\left(\frac{1}{\mu\gamma\Delta_i^U} \log n\right) .$$

The regret of any suboptimal column $j \in [L]$ is bounded analogously. □



(a) (b) (c)

Figure 7.1: The n-step regret of `Rank1ElimKL`, `UCB1Elim`, `Rank1Elim` and `UCB1` on problem (7.8) for (a) $K = L = 32$ (b) $K = L = 64$ (c) $K = L = 128$. The results are averaged over 20 runs.

7.5 Experiments

We conduct two experiments. In Section 7.5.1, we compare our algorithm to other algorithms in the literature on a synthetic problem. In Section 7.5.2, we evaluate the same algorithms on click models that are trained on a real-world dataset.

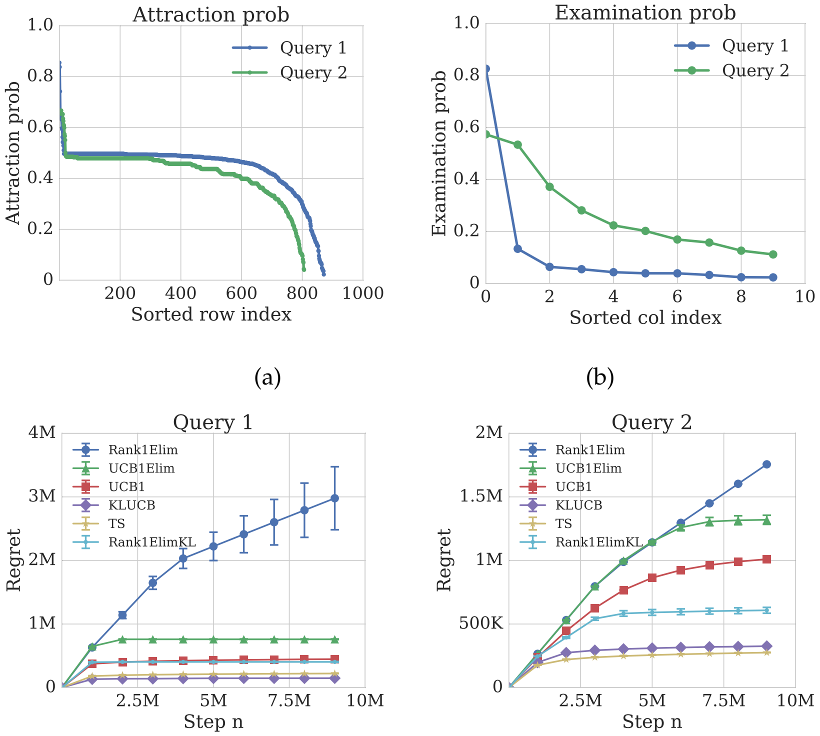
7.5.1 Comparison to Alternative Algorithms

Following Katariya *et al.* Katariya et al. [2017], we consider the "needle in a haystack" class of problems, where only one item is attractive and one position is examined. We recall the problem here. The i-th entry of $\mathbf{u}_t$, $\mathbf{u}_t(i)$, and the j-th entry of $\mathbf{v}_t$, $\mathbf{v}_t(j)$, are independent Bernoulli variables with means

$$\begin{aligned} \bar{u}(i) &= p_u + \Delta_u \mathbb{1}\{i = 1\}, \\ \bar{v}(j) &= p_v + \Delta_v \mathbb{1}\{j = 1\}, \end{aligned} \tag{7.8}$$

for some $(p_u, p_v) \in [0, 1]^2$ and gaps $(\Delta_u, \Delta_v) \in (0, 1 - p_u] \times (0, 1 - p_v]$. Note that arm $(1, 1)$ is optimal with an expected reward of $(p_u + \Delta_u)(p_v + \Delta_v)$.

The goal of this experiment is to compare `Rank1ElimKL` with five other algorithms from the literature and validate that its regret scales linearly with K and L, which implies that it exploits the problem structure. In this experiment, we set



(a) (b)

(c) (d)

Figure 7.2: (a) The sorted attraction probabilities of the items from 2 queries from the Yandex dataset. (b) The sorted examination probabilities of the positions for the same 2 queries. (c) The n-step regret in Query 1. (d) The n-step regret in Query 2. The results are averaged over 5 runs.

$p_U = p_V = 0.25, \Delta_U = \Delta_V = 0.5$, and $K = L$, so that $\mu = (1 - 1/K)0.25 + 0.75/K = 0.25 + 0.5/K$, $1 - p_{\max} = 0.25$, and $\gamma = \mu = 0.25 + 0.5/K$.

In addition to comparing to `Rank1Elim`, we also compare to `UCB1Elim` Auer and Ortner [2010], `UCB1` Auer et al. [2002a], `KL-UCB` Garivier and Cappe [2011], and Thompson sampling Thompson [1933]. `UCB1` is chosen as a baseline as it has been used by Katariya *et al.* Katariya et al. [2017] in their experiments. `UCB1Elim` uses an elimination approach similar to `Rank1Elim` and `Rank1ElimKL`. `KL-UCB` is similar to `UCB1`, but it uses `KL-UCB` confidence intervals. Thompson sampling (`TS`) is a Bayesian algorithm that maximizes the expected reward with respect to a randomly drawn belief.

Fig. 7.1 shows the n-step regret of the algorithms described above as a function of time n for $K = L$, the latter of which doubles from one plot to the next. We observe that the regret of `Rank1ElimKL` flattens in all three problems, which indicates that `Rank1ElimKL` learns the optimal arm. We also see that the regret of `Rank1ElimKL` doubles as K and L double, indicating that our bound in Theorem 7.1 has the right scaling in $K + L$, and that the algorithm leverages the problem structure. On the other hand, the regret of `UCB1`, `UCB1Elim`, `KL-UCB` and `TS` quadruples when K and L double, confirming that their regret is $\Omega(KL)$. Next, we observe that while `KL-UCB` and `TS` have smaller regret than `Rank1ElimKL` when K and L are small, the $(K + L)$-scaling of `Rank1ElimKL` enables it to outperform these algorithms for large K and L (Fig. 7.1c). Finally, note that `Rank1ElimKL` outperforms `Rank1Elim` in all three experiments, confirming the importance of tighter confidence intervals. It is worth noting that $\mu = \gamma$ for this problem, and hence $\mu^2 = \mu\gamma$. According to Theorem 7.1, `Rank1ElimKL` should not perform better than `Rank1Elim`. Yet it is 4 times better as seen in Fig. 7.1a. This suggests that our upper bound is loose.

7.5.2 Models Based on Real-World Data

In this experiment, we compare `Rank1ElimKL` to other algorithms on click models that are trained on the *Yandex* dataset Yandex, an anonymized search log of 35M search sessions. Each session contains a query, the list of displayed documents at positions 1 to 10, and the clicks on those documents. We select 20 most frequent queries from the dataset, and estimate the parameters of the PBM model using the EM algorithm Markov [2014], Chuklin et al. [2015b].

To illustrate our learned models, we plot the parameters of two queries, Queries 1 and 2. Fig. 7.2a shows the sorted attraction probabilities of items in the queries, and Fig. 7.2b shows the sorted examination probabilities of the positions. Query 1 has $L = 871$ items and Query 2 has $L = 807$ items. $K = 10$ is the number of documents displayed per query. We illustrate the performance on these queries because they differ notably in their μ (7.3) and $p_{\max}$ (7.4), so we can study the performance of our algorithm in different real-world settings. Fig. 7.2c and d show the regret of all algorithms on Queries 1 and 2, respectively.

We first note that KL-UCB and TS do better than Rank1ElimKL on both queries. As seen in Section 7.5.1, Rank1ElimKL is expected to improve over these baselines for large K and L, which is not the case here. With respect to other algorithms, we see that Rank1ElimKL is significantly better than Rank1Elim and UCB1Elim and no worse than UCB1 on Query 1, while for Query 2, Rank1ElimKL is superior to all of them. Note that $p_{\max} = 0.85$ in Query 1 is higher than $p_{\max} = 0.66$ in Query 2. Also, $\mu = 0.13$ in Query 1 is lower than $\mu = 0.28$ in Query 2. From (7.5), $\gamma = 0.15$ in Query 1, which is lower than $\gamma = 0.34$ in Query 2. Our upper bound (Theorem 7.1) on the regret of Rank1ElimKL scales as $\mathcal{O}((\mu\gamma)^{-1})$, and so we expect Rank1ElimKL to perform better on Query 2. Our results confirm this expectation.

In Fig. 7.3, we plot the average regret over all 20 queries, where the standard error is computed by repeating this procedure 5 times. Rank1ElimKL has the lowest regret of all algorithms except for KL-UCB and TS. Its regret is 10.9 percent lower than that of UCB1, and 79 percent lower than that of Rank1Elim. This is expected. Some real-world instances have a benign rank-1 structure like Query 2, while others do not, like Query 1. Hence, we see a reduction in the average gains of Rank1ElimKL over UCB1 in Fig. 7.3 as compared to Fig. 7.2d. The high regret of Rank1Elim, which is also designed to exploit the problem structure, shows that it is more sensitive to unfavorable rank-1 structures. Thus, the good news is that Rank1ElimKL improves on this limitation of Rank1Elim. However, KL-UCB and TS perform better on average, and we believe this is due to the fact 14 out of our 20 queries have $L < 200$, and hence $KL < 2000$. This is in line with the results of Section 7.5.1, which suggest that the advantage of Rank1ElimKL over KL-UCB and TS will "kick in" only for much larger values of K and L.

7.6 Related Work

Our algorithm is based on Rank1Elim of Katariya *et al.* Katariya et al. [2017]. The main difference is that we replace the confidence intervals of Rank1Elim, which

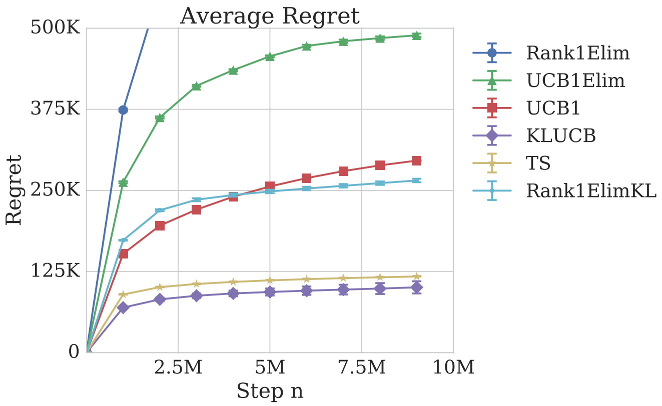


Figure 7.3: The average n-step regret over all 20 queries from the Yandex dataset, with 5 runs per query.

are based on subgaussian tail inequalities, with confidence intervals based on KL divergences. As discussed beforehand, this results in an unilateral improvement of their regret bound. The new algorithm is still able to exploit the problem structure of benign instances, while its regret is controlled on problem instances that are "hard" for `Rank1Elim`. As demonstrated in the previous section, the new algorithm is also a major practical improvement over `Rank1Elim`, while it remains competitive with alternatives on hard instances.

Several other papers studied bandits where the payoff is given by a low rank matrix. Zhao *et al.* Zhao et al. [2013] proposed a bandit algorithm for low-rank matrix completion, which approximates the posterior over latent item features by a single point. The authors do not analyze this algorithm. Kawale *et al.* Kawale et al. [2015] proposed a bandit algorithm for low-rank matrix completion using Thompson sampling with Rao-Blackwellization. They analyze a variant of their algorithm whose n-step regret for rank-1 matrices is $O((1/\Delta^2)\log n)$. This is suboptimal compared to our algorithm. Maillard *et al.* Maillard and Mannor [2014] studied a bandit problem where the arms are partitioned into latent groups. In this work, we do not make any such assumptions, but our results are limited to rank 1. Gentile *et al.* Gentile et al. [2014] proposed an algorithm that clusters users based on their preferences, under the assumption that the features of items are known. Sen *et al.* Sen et al. [2017] proposed an algorithm for contextual bandits with latent confounders, which reduces to a multi-armed bandit problem where the reward matrix is low-rank. They use an NMF-based approach and require that the reward matrix obeys a variant of the restricted isometry property. We make no such assumptions. Also, our learning agent controls both the row and column while in the above papers, the rows are controlled by the environment.

`Rank1ElimKL` is motivated by the structure of the PBM Richardson et al. [2007]. Lagree *et al.* Lagree et al. [2016] proposed a bandit algorithm for this model but they assume that the examination probabilities are known. `Rank1ElimKL` can be used to solve this problem without this assumption. The cascade model Craswell et al. [2008] is an alternative way of explaining the position bias in click data Chuklin et al. [2015b]. Bandit algorithms for this class of models have been proposed in several recent papers Kveton et al. [2015a], Combes et al. [2015a], Kveton et al. [2015b], Katariya et al. [2016], Zong et al. [2016], Li et al. [2016b].

7.7 Conclusions

In this work, we proposed `Rank1ElimKL`, an elimination algorithm that uses `KL-UCB` confidence intervals to find the maximum entry of a stochastic rank-1 matrix with Bernoulli rewards. The algorithm is a modification of `Rank1Elim` Katariya et al. [2017], where the subgaussian confidence intervals are replaced by the ones with KL divergences. As we demonstrate both empirically and analytically, this change results in a significant improvement. As a result, we obtain the first algorithm that is able to exploit the rank-1 structure without paying a significant penalty on instances where the rank-1 structure cannot be exploited.

We note that `Rank1ElimKL` uses the rank-1 structure of the problem and that there are no guarantees beyond rank-1. While the dependence of the regret of `Rank1ElimKL` on Δ is known to be tight Katariya et al. [2017], the question about the optimal dependence on μ is still open. Finally, we point out that `TS` and `KL-UCB` perform better than `Rank1ElimKL` in our experiments, especially for small $\mathtt{L}$ and $\mathtt{K}$. This is because `Rank1ElimKL` is an elimination algorithm. Elimination algorithms tend to have higher regret initially than UCB-style algorithms because they explore more aggressively. It is not inconceivable to have TS algorithms that leverage the rank-1 structure in the future.

7.8 Appendix

7.8.1 Proof of Theorem 7.1

We start by recalling Theorem 10 of Garivier and Cappe Garivier and Cappe [2011] with a slight extension that follows immediately by inspecting their proof. We will comment on the difference after stating the definitions. Let $(X_t)_{t\geqslant 1}$ be a sequence of random variables bounded in $[0,1]$. Assume that $(\mathcal{F}_t)_{t\geqslant 1}$ is a filtration ($\mathcal{F}_t \subset \mathcal{F}_{t+1}$ are σ-algebras) and $(X_t)_{t\geqslant 1}$ is $(\mathcal{F}_t)_t$-adapted (i.e., for $t \geqslant 1$, $X_1, \dots, X_t$ are $\mathcal{F}_t$ measurable), and $\mathbb{E}[X_{t+1}|\mathcal{F}_t] = \mu$ with some fixed value $\mu \in [0,1]$. Let $(\varepsilon_t)_{t\geqslant 1}$ be a sequence of $(\mathcal{F}_t)$-previsible Bernoulli random variables: For all $t \geqslant 1$, ε_t is $\mathcal{F}_{t-1}$-measurable with $\mathcal{F}_0 = \mathcal{F}$ the σ-algebra that holds all random variables. Define

$$S(t) = \sum_{s=1}^{t} \varepsilon_s X_s\,, \quad N(t) = \sum_{s=1}^{t} \varepsilon_s\,, \quad \hat{\mu}(t) = \frac{S(t)}{N(t)}\,, \quad t \geqslant 1\,.$$

The difference to the assumptions used by Garivier and Cappe Garivier and Cappe [2011] is that they assume that the random variables $(X_t)_{t\geqslant 1}$ are independent with common mean μ and that for $s > t$, X_s is independent of $\mathcal{F}_t$. With this we are ready to state their theorem:

Theorem 7.3 (After Theorem 10 of Garivier and Cappe Garivier and Cappe [2011]). *Let $(\hat{\mu}(t))_{t\geqslant 1}$ be as above and let*

$$U(t) = \sup\{\, q > \hat{\mu}(t) \,:\, N(t)\, d(\hat{\mu}(t), q) \leqslant \delta \,\}\,.$$

Then,

$$P(U(t) < \mu) \leqslant e \lceil \delta \log(t) \rceil \exp(-\delta)\,.$$

Let us now turn to our proof. Let $\mathbf{R}^{u}_{\ell}(i)$ be the stochastic regret associated with row i in row exploration stage ℓ and $\mathbf{R}^{v}_{\ell}(j)$ be the stochastic regret associated with column j in column exploration stage ℓ. Then the expected n-step regret of `Rank1ElimKL` can be written as

$$R(n) \leqslant \mathbb{E}\left[\sum_{\ell=0}^{n-1}\left(\sum_{i=1}^{K}\mathbf{R}^{u}_{\ell}(i) + \sum_{j=1}^{L}\mathbf{R}^{v}_{\ell}(j)\right)\right]\,,$$

where the outer sum is over possibly n stages. Let

$$\begin{aligned}\mathcal{E}^{u}_{\ell} = \{&\text{Event 1: } \forall i \in \mathbf{I}_{\ell} : \bar{\mathbf{u}}_{\ell}(i) \in [\mathbf{L}^{u}_{\ell}(i), \mathbf{U}^{u}_{\ell}(i)]\,,\\ &\text{Event 2: } \forall i \in \mathbf{I}_{\ell} : \bar{\mathbf{u}}_{\ell}(i) \geqslant \mu\bar{u}(i)\,,\\ &\text{Event 3: } \forall i \in \mathbf{I}_{\ell} \setminus \{i^*\} : n_{\ell} \geqslant \frac{16}{\mu\gamma(\Delta^{u}_{i})^2}\log n \implies \hat{\mathbf{u}}(i) \leqslant \mathbf{c}_{\ell}[\bar{u}(i) + \Delta^{u}_{i}/4]\,,\\ &\text{Event 4: } \forall i \in \mathbf{I}_{\ell} \setminus \{i^*\} : n_{\ell} \geqslant \frac{16}{\mu\gamma(\Delta^{u}_{i})^2}\log n \implies \hat{\mathbf{u}}(i^*) \geqslant \mathbf{c}_{\ell}[\bar{u}(i^*) - \Delta^{u}_{i}/4]\}\end{aligned}$$

be "good events" associated with row i at the end of stage ℓ, where

$$\bar{\mathbf{u}}_{\ell}(i) = \sum_{t=0}^{\ell}\mathbb{E}\left[\sum_{j=1}^{L}\frac{\mathbf{C}^{u}_{t}(i,j) - \mathbf{C}^{u}_{t-1}(i,j)}{n_{\ell}}\,\middle|\,\mathbf{h}^{v}_{t}\right] = \underbrace{\left(\sum_{t=0}^{\ell}\frac{n_t - n_{t-1}}{n_{\ell}}\sum_{j=1}^{L}\frac{\bar{v}(\mathbf{h}^{v}_{t}(j))}{L}\right)}_{\mathbf{c}_{\ell}}\bar{u}(i)$$

is the expected reward of row i conditioned on column elimination strategy $\mathbf{h}^{v}_{0}, \ldots, \mathbf{h}^{v}_{\ell}$;

$\mathbf{C}^{\mathsf{u}}_{-1}(i, j) = 0$; and $n_{-1} = 0$. Let $\overline{\mathcal{E}^{\mathsf{u}}_\ell}$ be the complement of event $\mathcal{E}^{\mathsf{u}}_\ell$. Let

$$\begin{aligned}\mathcal{E}^{\mathsf{v}}_\ell = \{&\text{Event 1: } \forall j \in \mathbf{J}_\ell : \bar{\mathbf{v}}_\ell(j) \in [\mathbf{L}^{\mathsf{v}}_\ell(j), \mathbf{U}^{\mathsf{v}}_\ell(j)]\,, \\ &\text{Event 2: } \forall j \in \mathbf{J}_\ell : \bar{\mathbf{v}}_\ell(j) \geqslant \mu\bar{v}(j)\,, \\ &\text{Event 3: } \forall j \in \mathbf{J}_\ell \setminus \{j^*\} : n_\ell \geqslant \frac{16}{\mu\gamma(\Delta^{\mathsf{v}}_j)^2}\log n \implies \hat{\mathbf{v}}(j) \leqslant \mathbf{c}_\ell[\bar{v}(j) + \Delta^{\mathsf{v}}_j/4]\,, \\ &\text{Event 4: } \forall j \in \mathbf{J}_\ell \setminus \{j^*\} : n_\ell \geqslant \frac{16}{\mu\gamma(\Delta^{\mathsf{v}}_j)^2}\log n \implies \hat{\mathbf{v}}(j^*) \geqslant \mathbf{c}_\ell[\bar{v}(j^*) - \Delta^{\mathsf{v}}_j/4]\}\end{aligned}$$

be "good events" associated with column j at the end of stage ℓ, where

$$\bar{\mathbf{v}}_\ell(j) = \sum_{t=0}^{\ell} \mathbb{E}\left[\sum_{i=1}^{K} \frac{\mathbf{C}^{\mathsf{v}}_t(i, j) - \mathbf{C}^{\mathsf{v}}_{t-1}(i, j)}{n_\ell} \,\middle|\, \mathbf{h}^{\mathsf{u}}_t\right] = \underbrace{\left(\sum_{t=0}^{\ell} \frac{n_t - n_{t-1}}{n_\ell} \sum_{i=1}^{K} \frac{\bar{u}(\mathbf{h}^{\mathsf{u}}_t(i))}{K}\right)}_{\mathbf{c}_\ell} \bar{v}(j)$$

is the expected reward of column j conditioned on row elimination strategy $\mathbf{h}^{\mathsf{u}}_0, \dots, \mathbf{h}^{\mathsf{u}}_\ell$; $\mathbf{C}^{\mathsf{v}}_{-1}(i, j) = 0$; and $n_{-1} = 0$. Let $\overline{\mathcal{E}^{\mathsf{v}}_\ell}$ be the complement of event $\mathcal{E}^{\mathsf{v}}_\ell$. Let $\mathcal{E}$ be the event that all events $\mathcal{E}^{\mathsf{u}}_\ell$ and $\mathcal{E}^{\mathsf{v}}_\ell$ happen; and $\overline{\mathcal{E}}$ be the complement of $\mathcal{E}$, the event that at least one of $\mathcal{E}^{\mathsf{u}}_\ell$ and $\mathcal{E}^{\mathsf{v}}_\ell$ does not happen. Then the expected n-step regret can be bounded from above as

$$\begin{aligned}R(n) &\leqslant \mathbb{E}\left[\left(\sum_{\ell=0}^{n-1}\left(\sum_{i=1}^{K} \mathbf{R}^{\mathsf{u}}_\ell(i) + \sum_{j=1}^{L} \mathbf{R}^{\mathsf{v}}_\ell(j)\right)\right)\mathbb{1}\{\mathcal{E}\}\right] + nP(\overline{\mathcal{E}}) \\ &\leqslant \mathbb{E}\left[\left(\sum_{\ell=0}^{n-1}\left(\sum_{i=1}^{K} \mathbf{R}^{\mathsf{u}}_\ell(i) + \sum_{j=1}^{L} \mathbf{R}^{\mathsf{v}}_\ell(j)\right)\right)\mathbb{1}\{\mathcal{E}\}\right] + (K + L)(6e + 2) \\ &= \sum_{i=1}^{K} \mathbb{E}\left[\sum_{\ell=0}^{n-1} \mathbf{R}^{\mathsf{u}}_\ell(i)\mathbb{1}\{\mathcal{E}\}\right] + \sum_{j=1}^{L} \mathbb{E}\left[\sum_{\ell=0}^{n-1} \mathbf{R}^{\mathsf{v}}_\ell(j)\mathbb{1}\{\mathcal{E}\}\right] + (K + L)(6e + 2)\,,\end{aligned}$$

where the second inequality is from Lemma 7.4.

Let $\mathcal{H}_\ell = (\mathbf{I}_\ell, \mathbf{J}_\ell)$ be the rows and columns in stage ℓ, and

$$\mathcal{F}_\ell = \left\{\forall i \in \mathbf{I}_\ell : \sqrt{\mu\gamma}\Delta^{\mathsf{u}}_i \leqslant \tilde{\Delta}_{\ell-1},\ \forall j \in \mathbf{J}_\ell : \sqrt{\mu\gamma}\Delta^{\mathsf{v}}_j \leqslant \tilde{\Delta}_{\ell-1}\right\}$$

be the event that all rows and columns with "large gaps" are eliminated by the beginning of stage ℓ. By Lemma 7.5, event $\mathcal{F}_\ell$ happens when event $\mathcal{E}$ happens. Moreover, the expected regret in stage ℓ is independent of $\mathcal{F}_\ell$ given $\mathcal{H}_\ell$. Therefore, we can bound the regret from above as

$$R(n) \leqslant \sum_{i=1}^{K} \mathbb{E}\left[\sum_{\ell=0}^{n-1} \mathbb{E}\left[\mathbf{R}_\ell^U(i) \,|\, \mathcal{H}_\ell\right] \mathbb{1}\{\mathcal{F}_\ell\}\right] + \sum_{j=1}^{L} \mathbb{E}\left[\sum_{\ell=0}^{n-1} \mathbb{E}\left[\mathbf{R}_\ell^V(j) \,|\, \mathcal{H}_\ell\right] \mathbb{1}\{\mathcal{F}_\ell\}\right] + (K+L)(6e+2)\,. \tag{7.9}$$

By Lemma 7.6,

$$\mathbb{E}\left[\sum_{\ell=0}^{n-1} \mathbb{E}\left[\mathbf{R}_\ell^U(i) \,|\, \mathcal{H}_\ell\right] \mathbb{1}\{\mathcal{F}_\ell\}\right] \leqslant \frac{160}{\mu\gamma\bar{\Delta}_i^U} \log n + 80\,,$$
$$\mathbb{E}\left[\sum_{\ell=0}^{n-1} \mathbb{E}\left[\mathbf{R}_\ell^V(j) \,|\, \mathcal{H}_\ell\right] \mathbb{1}\{\mathcal{F}_\ell\}\right] \leqslant \frac{160}{\mu\gamma\bar{\Delta}_j^V} \log n + 80\,.$$

Now we apply the above upper bounds to (7.9) and get our main claim.

7.8.2 Technical Lemmas

Lemma 7.4. *Let $\overline{\mathcal{E}}$ be defined as in the proof of Theorem 7.1. Then for any $n \geqslant 5$,*

$$P(\overline{\mathcal{E}}) \leqslant \frac{(K+L)(6e+2)}{n}\,.$$

Proof. Let $\mathcal{E}_\ell = \mathcal{E}_\ell^U \cap \mathcal{E}_\ell^V$. Then, $\overline{\mathcal{E}} = \overline{\mathcal{E}_0} \cup (\overline{\mathcal{E}_1} \cap \mathcal{E}_0) \cup \cdots \cup (\overline{\mathcal{E}_{n-1}} \cap \mathcal{E}_0 \cap \cdots \cap \mathcal{E}_{n-2})$. By the same logic, $\overline{\mathcal{E}_\ell} \cap \mathcal{E}_0 \cap \cdots \cap \mathcal{E}_{\ell-1} = (\overline{\mathcal{E}_\ell^U} \cap \mathcal{E}_0 \cap \cdots \cap \mathcal{E}_{\ell-1}) \cup (\overline{\mathcal{E}_\ell^V} \cap \mathcal{E}_\ell^U \cap \mathcal{E}_0 \cap \cdots \cap \mathcal{E}_{\ell-1})$. Hence,

$$P(\overline{\mathcal{E}}) \leqslant \sum_{\ell=0}^{n-1} P(\overline{\mathcal{E}_\ell^U}, \mathcal{E}_0, \ldots, \mathcal{E}_{\ell-1}) + P(\overline{\mathcal{E}_\ell^V}, \mathcal{E}_0, \ldots, \mathcal{E}_{\ell-1})\,.$$

Now we bound the probability of the events $\overline{\mathcal{E}_\ell^U}, \mathcal{E}_0^U, \ldots, \mathcal{E}_{\ell-1}^U, \mathcal{E}_0^V, \ldots, \mathcal{E}_{\ell-1}^V$; and then sum them up. The proof for the probability of the second term above is analogous and hence it is omitted.

Event 1

The probability that event 1 in $\mathcal{E}_\ell^u$ does not happen is bounded as follows. For any $i \in [K]$ and $\mathbf{h}_0^v, \dots, \mathbf{h}_\ell^v$,

$$\begin{aligned}
P(\bar{\mathbf{u}}_\ell(i) \notin [\mathbf{L}_\ell^u(i), \mathbf{U}_\ell^u(i)]) &\leqslant P(\bar{\mathbf{u}}_\ell(i) < \mathbf{L}_\ell^u(i)) + P(\bar{\mathbf{u}}_\ell(i) > \mathbf{U}_\ell^u(i)) \\
&\leqslant \frac{2e \left\lceil \log(n \log^3 n) \log n_\ell \right\rceil}{n \log^3 n} \\
&\leqslant \frac{2e \left\lceil \log^2 n + \log(\log^3 n) \log n \right\rceil}{n \log^3 n} \\
&\leqslant \frac{2e \left\lceil 2 \log^2 n \right\rceil}{n \log n} \\
&\leqslant \frac{6e}{n \log n},
\end{aligned}$$

where the second inequality is from Theorem 7.3, the third inequality is from $n \geqslant n_\ell$, the fourth inequality is from $\log(\log^3 n) \leqslant \log n$ for $n \geqslant 5$, and the last inequality is from $\left\lceil 2 \log^2 n \right\rceil \leqslant 3 \log^2 n$ for $n \geqslant 3$. By the union bound,

$$P(\exists i \in \mathbf{I}_\ell \text{ s.t. } \bar{\mathbf{u}}_\ell(i) \notin [\mathbf{L}_\ell^u(i), \mathbf{U}_\ell^u(i)]) \leqslant \frac{6eK}{n \log n}$$

for any $\mathbf{I}_\ell$ and $\mathbf{h}_0^v, \dots, \mathbf{h}_\ell^v$. Finally, we take the expectation over $\mathbf{I}_\ell$ and $\mathbf{h}_0^v, \dots, \mathbf{h}_\ell^v$; and have that the probability that event 1 in $\mathcal{E}_\ell^u$ does not happen at the end of stage ℓ is bounded as above.

Event 2

Event 2 in $\mathcal{E}_\ell^u$ is guaranteed to happen, $\bar{\mathbf{u}}_\ell(i) \geqslant \mu \bar{u}(i)$ for all $i \in \mathbf{I}_\ell$. This claim holds trivially when $\ell = 0$, because all columns in row elimination stage 0 are chosen with the same probability. When $\ell > 0$, all column confidence intervals up to stage ℓ hold because events $\mathcal{E}_0^v, \dots, \mathcal{E}_{\ell-1}^v$ happen. Therefore, by the design of `Rank1ElimKL`, any eliminated column j up to stage ℓ is substituted with column j' such that $\bar{v}(j') \geqslant \bar{v}(j)$. Since the columns in any row elimination stage are chosen randomly, $\bar{\mathbf{u}}_\ell(i) \geqslant \mu\bar{u}(i)$ for all $i \in \mathbf{I}_\ell$.

Event 3

The probability that event 3 in $\mathcal{E}_\ell^u$ does not happen is bounded as follows. If the event does not happen in row i, then

$$n_\ell \geqslant \frac{16}{\mu\gamma(\Delta_i^u)^2} \log n\,, \quad \hat{\mathbf{u}}(i) > \mathbf{c}_\ell[\bar{u}(i) + \Delta_i^u/4]\,.$$

From Hoeffding's inequality and $\mathbb{E}\left[\hat{\mathbf{u}}(i)\right] = \mathbf{c}_\ell\bar{u}(i)$, we have that

$$P(\hat{\mathbf{u}}(i) > \mathbf{c}_\ell[\bar{u}(i) + \Delta_i^u/4]) \leqslant \exp[-n_\ell d(\mathbf{c}_\ell[\bar{u}(i) + \Delta_i^u/4], \mathbf{c}_\ell\bar{u}(i))]\,.$$

From our scaling lemma (Lemma 7.2), the inequality $\mathbf{c}_\ell \geqslant \mu$ and the definition $\gamma = \max(\mu, 1 - p_{\max})$, we have that

$$\exp[-n_\ell d(\mathbf{c}_\ell[\bar{u}(i) + \Delta_i^u/4], \mathbf{c}_\ell\bar{u}(i))] \leqslant \exp[-n_\ell\, \mu\gamma\, (\Delta_i^u)^2/8]\,.$$

Finally, from our assumption on n_ℓ, we conclude that

$$\exp[-n_\ell\mu\gamma(\Delta_i^u)^2/8] \leqslant \exp[-2\log n] = \frac{1}{n^2}\,.$$

Now we chain all inequalities and observe that event 3 in $\mathcal{E}_\ell^u$ does not happen with probability of at most K/n^2 for any $\mathbf{I}_\ell$ and $\mathbf{h}_0^v, \dots, \mathbf{h}_\ell^v$. Finally, we take the expectation over $\mathbf{I}_\ell$ and $\mathbf{h}_0^v, \dots, \mathbf{h}_\ell^v$; and have that the probability that event 3 in $\mathcal{E}_\ell^u$ does not happen at the end of stage ℓ is at most K/n^2.

Event 4

The probability that event 4 in $\mathcal{E}_\ell^u$ does not happen can be bounded similarly to that of event 3. If the event does not happen in row i, then

$$n_\ell \geqslant \frac{16}{\mu\gamma(\Delta_i^u)^2}\log n\,, \quad \hat{\mathbf{u}}(i^*) < \mathbf{c}_\ell[\bar{u}(i^*) - \Delta_i^u/4]\,.$$

Then by the same reasoning as in event 3,

$$\begin{aligned} P(\hat{\mathbf{u}}(i^*) < \mathbf{c}_\ell[\bar{u}(i^*) - \Delta_i^u/4]) &\leqslant \exp[-n_\ell d(\mathbf{c}_\ell[\bar{u}(i^*) - \Delta_i^u/4], \mathbf{c}_\ell\bar{u}(i^*))] \\ &\leqslant \exp[-n_\ell\mu\gamma(\Delta_i^u)^2/8] \\ &\leqslant \exp[-2\log n] \\ &= \frac{1}{n^2}\,. \end{aligned}$$

This implies that event 4 in $\mathcal{E}_\ell^u$ does not happen with probability of at most K/n^2 for any $\mathbf{I}_\ell$ and $\mathbf{h}_0^v, \dots, \mathbf{h}_\ell^v$. Finally, we take the expectation over $\mathbf{I}_\ell$ and $\mathbf{h}_0^v, \dots, \mathbf{h}_\ell^v$; and have that the probability that event 4 in $\mathcal{E}_\ell^u$ does not happen at the end of stage ℓ is at most K/n^2.

Total probability

Note that the maximum number of stages in `Rank1ElimKL` is $\log n$. By the union bound, we get that

$$\begin{aligned} P(\overline{\mathcal{E}}) &\leqslant \left(\frac{6eK}{n\log n} + \frac{K}{n^2} + \frac{K}{n^2}\right)\log n + \left(\frac{6eL}{n\log n} + \frac{L}{n^2} + \frac{L}{n^2}\right)\log n \\ &\leqslant \frac{(K+L)(6e+2)}{n}\,. \end{aligned}$$

This concludes our proof. □

Lemma 7.5. *Let $n \geqslant 5$. Let event $\mathcal{E}$ happen and m be the first stage where $\tilde{\Delta}_m < \sqrt{\mu\gamma}\Delta_i^u$. Then row i must be eliminated by the end of stage m. Moreover, let m be the first stage where $\tilde{\Delta}_m < \sqrt{\mu\gamma}\Delta_j^v$. Then column j must be eliminated by the end of stage m.*

Proof. We only prove the first claim. The other claim is proved analogously.

From the definition of n_m and our assumption on $\tilde{\Delta}_m$,

$$n_m \geqslant \frac{16}{\tilde{\Delta}_m^2} \log n > \frac{16}{\mu\gamma(\Delta_i^u)^2} \log n\,. \tag{7.10}$$

Suppose that $\mathbf{U}_m^u(i) \geqslant \mathbf{c}_m[\bar{u}(i) + \Delta_i^u/2]$ happens. Then from this assumption, the definition of $\mathbf{U}_m^u(i)$, and event 3 in $\mathcal{E}_m^u$,

$$\begin{aligned} d(\hat{\mathbf{u}}(i), \mathbf{U}_m^u(i)) &\geqslant d^+(\hat{\mathbf{u}}(i), \mathbf{c}_m[\bar{u}(i) + \Delta_i^u/2]) \\ &\geqslant d(\mathbf{c}_m[\bar{u}(i) + \Delta_i^u/4], \mathbf{c}_m[\bar{u}(i) + \Delta_i^u/2])\,, \end{aligned}$$

where $d^+(p, q) = d(p, q)\mathbb{1}\{p \leqslant q\}$. From our scaling lemma (Lemma 7.2), the inequality $\mathbf{c}_\ell \geqslant \mu$ and the definition $\gamma = \max(\mu, 1 - p_{\max})$, we further have that

$$d(\mathbf{c}_m[\bar{u}(i) + \Delta_i^u/4], \mathbf{c}_m[\bar{u}(i) + \Delta_i^u/2]) \geqslant \frac{\mu\gamma\,(\Delta_i^u)^2}{8}\,.$$

From the definition of $\mathbf{U}_m^u(i)$, $n \geqslant 5$, and above inequalities,

$$n_m = \frac{\log n + 3\log\log n}{d(\hat{\mathbf{u}}(i), \mathbf{U}_m^u(i))} \leqslant \frac{2\log n}{d(\hat{\mathbf{u}}(i), \mathbf{U}_m^u(i))} \leqslant \frac{16\log n}{\mu\gamma(\Delta_i^u)^2}\,.$$

This contradicts to (7.10), and therefore it must be true that $\mathbf{U}_m^u(i) < \mathbf{c}_m[\bar{u}(i) + \Delta_i^u/2]$.

Now suppose that $\mathbf{L}_m^u(i^*) \leqslant \mathbf{c}_m[\bar{u}(i^*) - \Delta_i^u/2]$ happens. Then from this assumption, the definition of $\mathbf{L}_m^u(i^*)$, and event 4 in $\mathcal{E}_m^u$,

$$\begin{aligned} d(\hat{\mathbf{u}}(i^*), \mathbf{L}_m^u(i^*)) &\geqslant d^-(\hat{\mathbf{u}}(i^*), \mathbf{c}_m[\bar{u}(i^*) - \Delta_i^u/2]) \\ &\geqslant d(\mathbf{c}_m[\bar{u}(i^*) - \Delta_i^u/4], \mathbf{c}_m[\bar{u}(i^*) - \Delta_i^u/2])\,, \end{aligned}$$

where $d^-(p, q) = d(p, q)\mathbb{1}\{p \geqslant q\}$. From our scaling lemma (Lemma 7.2), the inequality $\mathbf{c}_\ell \geqslant \mu$ and the definition $\gamma = \max(\mu, 1 - p_{\max})$, we further have that

$$d(\mathbf{c}_m[\bar{u}(i^*) - \Delta_i^u/4], \mathbf{c}_m[\bar{u}(i^*) - \Delta_i^u/2]) \geqslant \frac{\mu\gamma\,(\Delta_i^u)^2}{8}\,.$$

From the definition of $\mathbf{L}^{\mathrm{u}}_m(i^*)$, $n \geqslant 5$, and above inequalities,

$$n_m = \frac{\log n + 3 \log\log n}{d(\hat{\mathbf{u}}(i^*), \mathbf{L}^{\mathrm{u}}_m(i^*))} \leqslant \frac{2 \log n}{d(\hat{\mathbf{u}}(i^*), \mathbf{L}^{\mathrm{u}}_m(i^*))} \leqslant \frac{16 \log n}{\mu\gamma(\Delta^{\mathrm{u}}_i)^2} .$$

This contradicts to (7.10), and therefore it must be true that $\mathbf{L}^{\mathrm{u}}_m(i^*) > \mathbf{c}_m[\bar{u}(i^*) - \Delta^{\mathrm{u}}_i/2]$.

Finally, it follows that row i is eliminated by the end of stage m because

$$\mathbf{U}^{\mathrm{u}}_m(i) < \mathbf{c}_m[\bar{u}(i) + \Delta^{\mathrm{u}}_i/2] = \mathbf{c}_m[\bar{u}(i^*) - \Delta^{\mathrm{u}}_i/2] < \mathbf{L}^{\mathrm{u}}_m(i^*) .$$

This concludes our proof. □

Lemma 7.6. *The expected regret associated with any row* $i \in [K]$ *is bounded as*

$$\mathbb{E}\left[\sum_{\ell=0}^{n-1} \mathbb{E}\left[\mathbf{R}^{\mathrm{u}}_\ell(i) \,|\, \mathcal{H}_\ell\right] \mathbb{1}\{\mathcal{F}_\ell\}\right] \leqslant \frac{160}{\mu\gamma\bar{\Delta}^{\mathrm{u}}_i} \log n + 80 .$$

Moreover, the expected regret associated with any column $j \in [L]$ *is bounded as*

$$\mathbb{E}\left[\sum_{\ell=0}^{n-1} \mathbb{E}\left[\mathbf{R}^{\mathrm{v}}_\ell(j) \,|\, \mathcal{H}_\ell\right] \mathbb{1}\{\mathcal{F}_\ell\}\right] \leqslant \frac{160}{\mu\gamma\bar{\Delta}^{\mathrm{v}}_j} \log n + 80 .$$

Proof. We only prove the first claim. The other claim is proved analogously.

This proof has two parts. In the first part, we assume that row i is suboptimal. In the second part, we assume that row i is optimal, $\Delta^{\mathrm{u}}_i = 0$.

Row i is suboptimal

Let row i be suboptimal and m be the first stage where $\tilde{\Delta}_m < \sqrt{\mu\gamma}\Delta^{\mathrm{u}}_i$. Then row i is guaranteed to be eliminated by the end of stage m (Lemma 7.5), and therefore

$$\mathbb{E}\left[\sum_{\ell=0}^{n-1} \mathbb{E}\left[\mathbf{R}^{\mathrm{u}}_\ell(i) \,|\, \mathcal{H}_\ell\right] \mathbb{1}\{\mathcal{F}_\ell\}\right] \leqslant \mathbb{E}\left[\sum_{\ell=0}^{m} \mathbb{E}\left[\mathbf{R}^{\mathrm{u}}_\ell(i) \,|\, \mathcal{H}_\ell\right] \mathbb{1}\{\mathcal{F}_\ell\}\right] .$$

By Lemma 4 of Katariya *et al.* Katariya et al. [2017], the expected regret of choosing row i in stage ℓ can be bounded from above as

$$\mathbb{E}\left[\mathbf{R}_\ell^u(i) \,|\, \mathcal{H}_\ell\right] \mathbb{1}\{\mathcal{F}_\ell\} \leqslant (\Delta_i^u + 2^{m-\ell+1}\Delta_i^u)(n_\ell - n_{\ell-1})\,,$$

where n_ℓ is the number of steps by the end of stage ℓ, $2^{m-\ell+1}\Delta_i^u$ is an upper bound on the gap of any non-eliminated column in stage $\ell \leqslant m$, and $n_{-1} = 0$. The bound follows from the observation that if column j is not eliminated before stage ℓ, then

$$\Delta_j^v \leqslant \frac{\tilde{\Delta}_{\ell-1}}{\sqrt{\mu\gamma}} = \frac{2^{m-\ell+1}\tilde{\Delta}_m}{\sqrt{\mu\gamma}} < 2^{m-\ell+1}\Delta_i^u\,.$$

It follows that

$$\begin{aligned}
\sum_{\ell=0}^{m}(\Delta_i^u + 2^{m-\ell+1}\Delta_i^u)(n_\ell - n_{\ell-1}) &\leqslant \Delta_i^u n_m + \Delta_i^u \sum_{\ell=0}^{m} 2^{m-\ell+1} n_\ell \\
&\leqslant 2^4\Delta_i^u(2^{2m}\log n + 1) + 2^4\Delta_i^u \sum_{\ell=0}^{m} 2^{m-\ell+1}(2^{2m}\log n + 1) \\
&= 2^{2m+4}\Delta_i^u \log n + 16\Delta_i^u + 2^{2m+6}\Delta_i^u \log n + 64\Delta_i^u \\
&\leqslant 5 \cdot 2^6 \cdot 2^{2m-2}\Delta_i^u \log n + 80\,.
\end{aligned}$$

From the definition of m, we have that

$$2^{m-1} = \frac{1}{\tilde{\Delta}_{m-1}} \leqslant \frac{1}{\sqrt{\mu\gamma}\Delta_i^u}\,.$$

Now we chain all above inequalities and get that

$$\begin{aligned}
\mathbb{E}\left[\sum_{\ell=0}^{n-1} \mathbb{E}\left[\mathbf{R}_\ell^u(i) \,|\, \mathcal{H}_\ell\right] \mathbb{1}\{\mathcal{F}_\ell\}\right] &\leqslant \sum_{\ell=0}^{m}(\Delta_i^u + 2^{m-\ell+1}\Delta_i^u)(n_\ell - n_{\ell-1}) \\
&\leqslant \frac{160}{\mu\gamma\Delta_i^u}\log n + 80\,.
\end{aligned}$$

This concludes the first part of our proof.

Row i is optimal

Let row i be optimal and m be the first stage where $\tilde{\Delta}_m < \sqrt{\mu\gamma}\Delta^{v}_{\min}$. Then similarly to the first part of the analysis,

$$\mathbb{E}\left[\sum_{\ell=0}^{n-1} \mathbb{E}\left[\mathbf{R}^{u}_{\ell}(i) \,|\, \mathcal{H}_\ell\right] \mathbb{1}\{\mathcal{F}_\ell\}\right] \leqslant \frac{160}{\mu\gamma\Delta^{v}_{\min}} \log n + 80\,.$$

This concludes our proof. □

Algorithm 7 `Rank1ElimKL` for Bernoulli rank-1 bandits.

```
// Initialization
$t \leftarrow 1,\ \tilde{\Delta}_0 \leftarrow 1, n_{-1} \leftarrow 0$
$\mathbf{C}_0^{\mathrm{u}} \leftarrow 0_{K,L}, \mathbf{C}_0^{\mathrm{v}} \leftarrow 0_{K,L}$ // Zero matrix with $K$ rows and $L$ columns
$\mathbf{h}_0^{\mathrm{u}} \leftarrow (1, \dots, K),\ \mathbf{h}_0^{\mathrm{v}} \leftarrow (1, \dots, L)$

for $\ell = 0, 1, \dots$ do
    $n_\ell \leftarrow \lceil 16 \tilde{\Delta}_\ell^{-2} \log n \rceil$
    $\mathbf{I}_\ell \leftarrow \bigcup_{i \in [K]} \{\mathbf{h}_\ell^{\mathrm{u}}(i)\},\ \mathbf{J}_\ell \leftarrow \bigcup_{j \in [L]} \{\mathbf{h}_\ell^{\mathrm{v}}(j)\}$

    // Row and column exploration
    for $n_\ell - n_{\ell-1}$ times do
        Choose uniformly at random column $j \in [L]$
        $j \leftarrow \mathbf{h}_\ell^{\mathrm{v}}(j)$
        for all $i \in \mathbf{I}_\ell$ do
            $\mathbf{C}_\ell^{\mathrm{u}}(i, j) \leftarrow \mathbf{C}_\ell^{\mathrm{u}}(i, j) + \mathbf{u}_t(i)\mathbf{v}_t(j)$
            $t \leftarrow t + 1$
        end for
        Choose uniformly at random row $i \in [K]$
        $i \leftarrow \mathbf{h}_\ell^{\mathrm{u}}(i)$
        for all $j \in \mathbf{J}_\ell$ do
            $\mathbf{C}_\ell^{\mathrm{v}}(i, j) \leftarrow \mathbf{C}_\ell^{\mathrm{v}}(i, j) + \mathbf{u}_t(i)\mathbf{v}_t(j)$
            $t \leftarrow t + 1$
        end for
    end for

    // UCBs and LCBs on the expected rewards of all remaining rows and columns with divergence constraint $\delta_\ell \leftarrow \log n + 3 \log \log n$

    for all $i \in \mathbf{I}_\ell$ do
        $\hat{\mathbf{u}}_\ell(i) \leftarrow n_\ell^{-1} \sum_{j=1}^{L} \mathbf{C}_\ell^{\mathrm{u}}(i, j)$
        $\mathbf{U}_\ell^{\mathrm{u}}(i) \leftarrow \arg\max_{q \in [\hat{\mathbf{u}}_\ell(i), 1]} \{n_\ell d(\hat{\mathbf{u}}_\ell(i), q) \leqslant \delta_\ell\}$
        $\mathbf{L}_\ell^{\mathrm{u}}(i) \leftarrow \arg\min_{q \in [0, \hat{\mathbf{u}}_\ell(i)]} \{n_\ell d(\hat{\mathbf{u}}_\ell(i), q) \leqslant \delta_\ell\}$
    end for
    for all $j \in \mathbf{J}_\ell$ do
        $\hat{\mathbf{v}}_\ell(j) \leftarrow n_\ell^{-1} \sum_{i=1}^{K} \mathbf{C}_\ell^{\mathrm{v}}(i, j)$
        $\mathbf{U}_\ell^{\mathrm{v}}(j) \leftarrow \arg\max_{q \in [\hat{\mathbf{v}}_\ell(j), 1]} \{n_\ell d(\hat{\mathbf{v}}_\ell(j), q) \leqslant \delta_\ell\}$
        $\mathbf{L}_\ell^{\mathrm{v}}(j) \leftarrow \arg\min_{q \in [0, \hat{\mathbf{v}}_\ell(j)]} \{n_\ell d(\hat{\mathbf{v}}_\ell(j), q) \leqslant \delta_\ell\}$
    end for

    // Row and column elimination
    $\mathbf{i}_\ell \leftarrow \arg\max_{i \in \mathbf{I}_\ell} \mathbf{L}_\ell^{\mathrm{u}}(i)$
    $\mathbf{h}_{\ell+1}^{\mathrm{u}} \leftarrow \mathbf{h}_\ell^{\mathrm{u}}$
    for $i = 1, \dots, K$ do
        if $\mathbf{U}_\ell^{\mathrm{u}}(\mathbf{h}_\ell^{\mathrm{u}}(i)) \leqslant \mathbf{L}_\ell^{\mathrm{u}}(\mathbf{i}_\ell)$ then
            $\mathbf{h}_{\ell+1}^{\mathrm{u}}(i) \leftarrow \mathbf{i}_\ell$
        end if
    end for

    $\mathbf{j}_\ell \leftarrow \arg\max_{j \in \mathbf{J}_\ell} \mathbf{L}_\ell^{\mathrm{v}}(j)$
    $\mathbf{h}_{\ell+1}^{\mathrm{v}} \leftarrow \mathbf{h}_\ell^{\mathrm{v}}$
    for $j = 1, \dots, L$ do
        if $\mathbf{U}_\ell^{\mathrm{v}}(\mathbf{h}_\ell^{\mathrm{v}}(j)) \leqslant \mathbf{L}_\ell^{\mathrm{v}}(\mathbf{j}_\ell)$ then
            $\mathbf{h}_{\ell+1}^{\mathrm{v}}(j) \leftarrow \mathbf{j}_\ell$
        end if
    end for

    $\tilde{\Delta}_{\ell+1} \leftarrow \tilde{\Delta}_\ell / 2,\ \mathbf{C}_{\ell+1}^{\mathrm{u}} \leftarrow \mathbf{C}_\ell^{\mathrm{u}},\ \mathbf{C}_{\ell+1}^{\mathrm{v}} \leftarrow \mathbf{C}_\ell^{\mathrm{v}}$
end for
```

BIBLIOGRAPHY

Yasin Abbasi-Yadkori, David Pal, and Csaba Szepesvari. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems 24*, pages 2312–2320, 2011.

Gediminas Adomavicius and Alexander Tuzhilin. Context-aware recommender systems. In *Recommender systems handbook*, pages 191–226. Springer, 2015.

Arpit Agarwal, Shivani Agarwal, Sepehr Assadi, and Sanjeev Khanna. Learning with limited rounds of adaptivity: Coin tossing, multi-armed bandits, and ranking from pairwise comparisons. In *Conference on Learning Theory*, pages 39–75, 2017.

Shivani Agarwal. On ranking and choice models. In *IJCAI*, pages 4050–4053, 2016.

Eugene Agichtein, Eric Brill, and Susan Dumais. Improving web search ranking by incorporating user behavior information. In *Proceedings of the 29th Annual International ACM SIGIR Conference*, pages 19–26, 2006.

Rajeev Agrawal, Demosthenis Teneketzis, and Venkatachalam Anantharam. Asymptotically efficient adaptive allocation schemes for controlled i.i.d. processes: Finite parameter space. *IEEE Transactions on Automatic Control*, 34(3): 258–267, 1989.

Nir Ailon. An active learning algorithm for ranking from pairwise preferences with an almost optimal query complexity. *Journal of Machine Learning Research*, 13 (Jan):137–164, 2012.

Nir Ailon, Moses Charikar, and Alantha Newman. Aggregating inconsistent information: ranking and clustering. *Journal of the ACM (JACM)*, 55(5):23, 2008.

Nir Ailon, Thorsten Joachims, and Zohar Karnin. Reducing dueling bandits to cardinal bandits. *arXiv preprint arXiv:1405.3396*, 2014.

Laurent Alonso, Philippe Chassaing, Florent Gillet, Svante Janson, Edward M Reingold, and Rene Schott. Sorting with unreliable comparisons: A probabilistic analysis. 2003.

Jean-Yves Audibert and Sébastien Bubeck. Best arm identification in multi-armed bandits. In *COLT-23th Conference on Learning Theory-2010*, pages 13–p, 2010.

Peter Auer. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 3(Nov):397–422, 2002.

Peter Auer and Ronald Ortner. UCB revisited: Improved regret bounds for the stochastic multi-armed bandit problem. *Periodica Mathematica Hungarica*, 61(1-2): 55–65, 2010.

Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47:235–256, 2002a.

Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3):235–256, 2002b.

G. Bartók and Cs. Szepesvári. Partial monitoring with side information. In *ALT*, pages 305–319, October 2012.

G. Bartók, D. Foster, D. Pál, A. Rakhlin, and Cs. Szepesvári. Partial monitoring – classification, regret bounds, and algorithms. *Mathematics of Operations Research*, 39:967–997, 2014.

Gabor Bartok, Navid Zolghadr, and Csaba Szepesvari. An adaptive algorithm for finite stochastic partial monitoring. In *Proceedings of the 29th International Conference on Machine Learning*, 2012.

Hamsa Bastani, Mohsen Bayati, and Khashayar Khosravi. Exploiting the natural exploration in contextual bandits. *arXiv preprint arXiv:1704.09011*, 2017.

Hila Becker, Christopher Meek, and David Maxwell Chickering. Modeling contextual factors of click rates. In *Proceedings of the 22nd AAAI Conference on Artificial Intelligence*, pages 1310–1315, 2007.

Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. *Concentration inequalities: A nonasymptotic theory of independence*. Oxford University Press, 2013.

Ralph Allan Bradley and Milton E Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

Mark Braverman and Elchanan Mossel. Sorting from noisy information. *arXiv preprint arXiv:0910.1191*, 2009.

Andrei Z Broder. Computational advertising and recommender systems. In *Proceedings of the 2008 ACM conference on Recommender systems*, pages 1–2. ACM, 2008.

Sébastian Bubeck, Tengyao Wang, and Nitin Viswanathan. Multiple identifications in multi-armed bandits. In *International Conference on Machine Learning*, pages 258–265, 2013.

Sébastien Bubeck, Rémi Munos, and Gilles Stoltz. Pure exploration in finitely-armed and continuous-armed bandits. *Theoretical Computer Science*, 412(19):1832–1852, 2011.

Sébastien Bubeck, Nicolo Cesa-Bianchi, et al. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning*, 5(1):1–122, 2012.

Róbert Busa-Fekete and Eyke Hüllermeier. A survey of preference-based online learning with bandit algorithms. In *International Conference on Algorithmic Learning Theory*, pages 18–39. Springer, 2014.

Olivier Cappe and Eric Moulines. Online EM algorithm for latent data models. *Journal of the Royal Statistical Society Series B*, 71(3):593–613, 2009.

Olivier Chapelle and Ya Zhang. A dynamic bayesian network click model for web search ranking. In *Proceedings of the 18th International Conference on World Wide Web*, pages 1–10, 2009.

Olivier Chapelle, Thorsten Joachims, Filip Radlinski, and Yisong Yue. Large-scale validation and analysis of interleaved search evaluation. *ACM Transactions on Information Systems (TOIS)*, 30(1):6, 2012.

Lijie Chen, Jian Li, and Mingda Qiao. Nearly instance optimal sample complexity bounds for top-k arm selection. In *Artificial Intelligence and Statistics*, pages 101–110, 2017.

Wei Chen, Yajun Wang, and Yang Yuan. Combinatorial multi-armed bandit: General framework, results and applications. In *Proceedings of the 30th International Conference on Machine Learning*, pages 151–159, 2013a.

Wei Chen, Yajun Wang, and Yang Yuan. Combinatorial multi-armed bandit: General framework and applications. In *International Conference on Machine Learning*, pages 151–159, 2013b.

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

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

Aleksandr Chuklin, Ilya Markov, and Maarten de Rijke. Click models for web search. *Synthesis Lectures on Information Concepts, Retrieval, and Services*, 7(3):1–115, 2015c.

Richard Combes, Stefan Magureanu, Alexandre Proutiere, and Cyrille Laroche. Learning to rank: Regret lower bounds and efficient algorithms. In *Proceedings of the 2015 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, 2015a.

Richard Combes, Mohammad Sadegh Talebi, Alexandre Proutiere, and Marc Lelarge. Combinatorial bandits revisited. In *Advances in Neural Information Processing Systems 28*, pages 2107–2115, 2015b.

Thomas M Cover and Joy A Thomas. *Elements of information theory*. John Wiley & Sons, 2012.

Nick Craswell, Onno Zoeter, Michael Taylor, and Bill Ramsey. An experimental comparison of click position-bias models. In *Proceedings of the 1st ACM International Conference on Web Search and Data Mining*, pages 87–94, 2008.

Paolo Cremonesi, Franca Garzotto, Sara Negro, Alessandro Vittorio Papadopoulos, and Roberto Turrin. Looking for "good" recommendations: A comparative evaluation of recommender systems. In *IFIP Conference on Human-Computer Interaction*, pages 152–168. Springer, 2011.

Varsha Dani, Thomas Hayes, and Sham Kakade. Stochastic linear optimization under bandit feedback. In *Proceedings of the 21st Annual Conference on Learning Theory*, pages 355–366, 2008.

Mukund Deshpande and George Karypis. Item-based top-n recommendation algorithms. *ACM Transactions on Information Systems (TOIS)*, 22(1):143–177, 2004.

Abhimanyu Dubey, Nikhil Naik, Devi Parikh, Ramesh Raskar, and César A Hidalgo. Deep learning the city: Quantifying urban perception at a global scale. In *European Conference on Computer Vision*, pages 196–212. Springer, 2016.

Jack Edmonds. Matroids and the greedy algorithm. *Mathematical programming*, 1 (1):127–136, 1971.

Eyal Even-Dar, Shie Mannor, and Yishay Mansour. Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems. *Journal of machine learning research*, 7(Jun):1079–1105, 2006.

Moein Falahatgar, Alon Orlitsky, Venkatadheeraj Pichapati, and Ananda Theertha Suresh. Maximum selection and ranking under noisy comparisons. *arXiv preprint arXiv:1705.05366*, 2017.

Uriel Feige, Prabhakar Raghavan, David Peleg, and Eli Upfal. Computing with noisy information. *SIAM Journal on Computing*, 23(5):1001–1018, 1994.

Sarah Filippi, Olivier Cappe, Aurelien Garivier, and Csaba Szepesvari. Parametric bandits: The generalized linear case. In *Advances in Neural Information Processing Systems 23*, pages 586–594, 2010.

Yi Gai, Bhaskar Krishnamachari, and Rahul Jain. Combinatorial network optimization with unknown variables: Multi-armed bandits with linear rewards and individual observations. *IEEE/ACM Transactions on Networking*, 20(5):1466–1478, 2012a.

Yi Gai, Bhaskar Krishnamachari, and Rahul Jain. Combinatorial network optimization with unknown variables: Multi-armed bandits with linear rewards and individual observations. *IEEE/ACM Transactions on Networking (TON)*, 20(5): 1466–1478, 2012b.

Aurélien Garivier. Informational confidence bounds for self-normalized averages and applications. In *Information Theory Workshop (ITW), 2013 IEEE*, pages 1–5. IEEE, 2013.

Aurelien Garivier and Olivier Cappe. The KL-UCB algorithm for bounded stochastic bandits and beyond. In *Proceeding of the 24th Annual Conference on Learning Theory*, pages 359–376, 2011.

Aurélien Garivier and Olivier Cappé. The kl-ucb algorithm for bounded stochastic bandits and beyond. In *COLT*, pages 359–376, 2011.

Claudio Gentile, Shuai Li, and Giovanni Zappella. Online clustering of bandits. In *Proceedings of the 31st International Conference on Machine Learning*, pages 757–765, 2014.

Todd Graves and Tze Leung Lai. Asymptotically efficient adaptive choice of control laws in controlled Markov chains. *SIAM Journal on Control and Optimization*, 35 (3):715–743, 1997.

Fan Guo, Chao Liu, Anitha Kannan, Tom Minka, Michael Taylor, Yi Min Wang, and Christos Faloutsos. Click chain model in web search. In *Proceedings of the 18th International Conference on World Wide Web*, pages 11–20, 2009a.

Fan Guo, Chao Liu, and Yi Min Wang. Efficient multiple-click models in web search. In *Proceedings of the 2nd ACM International Conference on Web Search and Data Mining*, pages 124–131, 2009b.

Reinhard Heckel, Nihar B Shah, Kannan Ramchandran, and Martin J Wainwright. Active ranking from pairwise comparisons and the futility of parametric assumptions. *arXiv preprint arXiv:1606.08842*, 2016.

Kevin Jamieson, Matthew Malloy, Robert Nowak, and Sébastien Bubeck. lil'ucb: An optimal exploration algorithm for multi-armed bandits. In *Conference on Learning Theory*, pages 423–439, 2014.

Kevin G Jamieson and Robert Nowak. Active ranking using pairwise comparisons. In *Advances in Neural Information Processing Systems*, pages 2240–2248, 2011.

Kevin G Jamieson, Lalit Jain, Chris Fernandez, Nicholas J Glattard, and Rob Nowak. Next: A system for real-world development, evaluation, and application of active learning. In *Advances in Neural Information Processing Systems*, pages 2656–2664, 2015a.

Kevin G Jamieson, Sumeet Katariya, Atul Deshpande, and Robert D Nowak. Sparse dueling bandits. In *AISTATS*, 2015b.

Haotian Jiang, Jian Li, and Mingda Qiao. Practical algorithms for best-k identification in multi-armed bandits. *arXiv preprint arXiv:1705.06894*, 2017.

Shivaram Kalyanakrishnan, Ambuj Tewari, Peter Auer, and Peter Stone. Pac subset selection in stochastic multi-armed bandits. In *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, pages 655–662, 2012.

Zohar Karnin, Tomer Koren, and Oren Somekh. Almost optimal exploration in multi-armed bandits. In *Proceedings of the 30th International Conference on Machine Learning*, 2013.

Sumeet Katariya, Branislav Kveton, Csaba Szepesvari, and Zheng Wen. DCM bandits: Learning to rank with multiple clicks. In *Proceedings of the 33rd International Conference on Machine Learning*, pages 1215–1224, 2016.

Sumeet Katariya, Branislav Kveton, Csaba Szepesvari, Claire Vernade, and Zheng Wen. Stochastic rank-1 bandits. In *AISTATS*, 2017.

Emilie Kaufmann and Shivaram Kalyanakrishnan. Information complexity in bandit subset selection. In *COLT*, pages 228–251, 2013.

Emilie Kaufmann, Olivier Cappé, and Aurélien Garivier. On the complexity of best arm identification in multi-armed bandit models. *arXiv preprint arXiv:1407.4443*, 2014.

Emilie Kaufmann, Olivier Cappé, and Aurélien Garivier. On the complexity of best arm identification in multi-armed bandit models. *The Journal of Machine Learning Research*, 2015.

Emilie Kaufmann, Olivier Cappe, and Aurelien Garivier. On the complexity of best-arm identification in multi-armed bandit models. *Journal of Machine Learning Research*, 17:1–42, 2016.

Jaya Kawale, Hung Bui, Branislav Kveton, Long Tran-Thanh, and Sanjay Chawla. Efficient Thompson sampling for online matrix-factorization recommendation. In *Advances in Neural Information Processing Systems 28*, pages 1297–1305, 2015.

Abbas Kazerouni, Mohammad Ghavamzadeh, Yasin Abbasi, and Benjamin Van Roy. Conservative contextual linear bandits. In *Advances in Neural Information Processing Systems*, pages 3913–3922, 2017.

Raghunandan Keshavan, Andrea Montanari, and Sewoong Oh. Matrix completion from noisy entries. *Journal of Machine Learning Research*, 11:2057–2078, 2010.

Tomas Kocak, Michal Valko, Remi Munos, and Shipra Agrawal. Spectral Thompson sampling. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence*, pages 1911–1917, 2014.

Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *IEEE Computer*, 42(8):30–37, 2009.

Branislav Kveton, Zheng Wen, Azin Ashkan, Hoda Eydgahi, and Brian Eriksson. Matroid bandits: Fast combinatorial optimization with learning. In *Proceedings of the 30th Conference on Uncertainty in Artificial Intelligence*, pages 420–429, 2014a.

Branislav Kveton, Zheng Wen, Azin Ashkan, Hoda Eydgahi, and Brian Eriksson. Matroid bandits: Fast combinatorial optimization with learning. *arXiv preprint arXiv:1403.5045*, 2014b.

Branislav Kveton, Csaba Szepesvari, Zheng Wen, and Azin Ashkan. Cascading bandits: Learning to rank in the cascade model. In *Proceedings of the 32nd International Conference on Machine Learning*, 2015a.

Branislav Kveton, Zheng Wen, Azin Ashkan, and Csaba Szepesvari. Combinatorial cascading bandits. In *Advances in Neural Information Processing Systems 28*, pages 1450–1458, 2015b.

Branislav Kveton, Zheng Wen, Azin Ashkan, and Csaba Szepesvari. Tight regret bounds for stochastic combinatorial semi-bandits. In *Proceedings of the 18th International Conference on Artificial Intelligence and Statistics*, 2015c.

Branislav Kveton, Zheng Wen, Azin Ashkan, and Csaba Szepesvari. Tight regret bounds for stochastic combinatorial semi-bandits. In *Artificial Intelligence and Statistics*, pages 535–543, 2015d.

Paul Lagree, Claire Vernade, and Olivier Cappe. Multiple-play bandits in the position-based model. In *Advances in Neural Information Processing Systems 29*, pages 1597–1605, 2016.

T. L. Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6(1):4–22, 1985.

Shyong Lam and Jon Herlocker. MovieLens 1M Dataset. http://www.grouplens.org/node/12, 2013.

Shuai Li, Alexandros Karatzoglou, and Claudio Gentile. Collaborative filtering bandits. In *Proceedings of the 39th Annual International ACM SIGIR Conference*, 2016a.

Shuai Li, Baoxiang Wang, Shengyu Zhang, and Wei Chen. Contextual combinatorial cascading bandits. In *Proceedings of the 33rd International Conference on Machine Learning*, pages 1245–1253, 2016b.

Tian Lin, Bruno Abrahao, Robert Kleinberg, John Lui, and Wei Chen. Combinatorial partial monitoring game with linear feedback and its applications. In *Proceedings of the 31st International Conference on Machine Learning*, pages 901–909, 2014.

Odalric-Ambrym Maillard and Shie Mannor. Latent bandits. In *Proceedings of the 31st International Conference on Machine Learning*, pages 136–144, 2014.

Shie Mannor and John N Tsitsiklis. The sample complexity of exploration in the multi-armed bandit problem. *The Journal of Machine Learning Research*, 5:623–648, 2004.

Ilya Markov. Pyclick - click models for web search. https://github.com/markovi/PyClick, 2014.

Lucas Maystre and Matthias Grossglauser. Just sort it! a simple and effective approach to active preference learning. In *Proceedings of Machine Learning Research*, number EPFL-CONF-229163, 2017.

Nikhil Naik, Jade Philipoom, Ramesh Raskar, and César Hidalgo. Streetscore-predicting the perceived safety of one million streetscapes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 779–785, 2014.

Sahand Negahban, Sewoong Oh, and Devavrat Shah. Iterative ranking from pair-wise comparisons. In *Advances in Neural Information Processing Systems*, pages 2474–2482, 2012a.

Sahand Negahban, Sewoong Oh, and Devavrat Shah. Rank centrality: Ranking from pair-wise comparisons. *arXiv preprint arXiv:1209.1688*, 2012b.

Tao Qin and Tie-Yan Liu. Introducing letor 4.0 datasets. *CoRR*, abs/1306.2597, 2013.

Tao Qin, Tie-Yan Liu, Jun Xu, and Hang Li. Letor: A benchmark collection for research on learning to rank for information retrieval. *Information Retrieval*, 13(4):346–374, 2010. ISSN 1386-4564. doi: 10.1007/s10791-009-9123-y. URL `http://dx.doi.org/10.1007/s10791-009-9123-y`.

Filip Radlinski and Thorsten Joachims. Query chains: Learning to rank from implicit feedback. In *Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 239–248, 2005.

Filip Radlinski and Thorsten Joachims. Minimally invasive randomization for collecting unbiased preferences from clickthrough. In *Logs, Proceedings of the 21st National Conference on Artificial Intelligence (AAAI*. Citeseer, 2006.

Filip Radlinski, Robert Kleinberg, and Thorsten Joachims. Learning diverse rankings with multi-armed bandits. In *Proceedings of the 25th International Conference on Machine Learning*, pages 784–791, 2008.

Maxim Raginsky and Igal Sason. Concentration of measure inequalities in information theory, communications and coding. *CoRR*, abs/1212.4663, 2012.

Arun Rajkumar and Shivani Agarwal. A statistical convergence perspective of algorithms for rank aggregation from pairwise data. In *Proceedings of the 31st International Conference on Machine Learning*, pages 118–126, 2014.

Arun Rajkumar, Suprovat Ghoshal, Lek-Heng Lim, and Shivani Agarwal. Ranking from stochastic pairwise preferences: Recovering condorcet winners and tournament solution sets at the top. In *International Conference on Machine Learning*, pages 665–673, 2015.

Paul Resnick and Hal R Varian. Recommender systems. *Communications of the ACM*, 40(3):56–58, 1997.

Matthew Richardson, Ewa Dominowska, and Robert Ragno. Predicting clicks: Estimating the click-through rate for new ads. In *Proceedings of the 16th International Conference on World Wide Web*, pages 521–530, 2007.

Tobias Schnabel, Paul N Bennett, Susan T Dumais, and Thorsten Joachims. Short-term satisfaction and long-term coverage: Understanding how users tolerate algorithmic exploration. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pages 513–521. ACM, 2018.

Robert Sedgewick and Kevin Wayne. *Algorithms*. Addison-Wesley Professional, 2011.

Rajat Sen, Karthikeyan Shanmugam, Murat Kocaoglu, Alexandros G Dimakis, and Sanjay Shakkottai. Contextual bandits with latent confounders: An nmf approach. In *AISTATS*, 2017.

Nihar Shah, Sivaraman Balakrishnan, Aditya Guntuboyina, and Martin Wainwright. Stochastically transitive models for pairwise comparisons: Statistical and computational issues. In *International Conference on Machine Learning*, pages 11–20, 2016.

Dan Siroker and Pete Koomen. *A/B testing: The most powerful way to turn clicks into customers*. John Wiley & Sons, 2013.

Aleksandrs Slivkins, Filip Radlinski, and Sreenivas Gollapudi. Ranked bandits in metric spaces: Learning diverse rankings over large document collections. *Journal of Machine Learning Research*, 14(1):399–436, 2013.

Mohammad Sadegh Talebi and Alexandre Proutiere. An optimal algorithm for stochastic matroid bandit optimization. In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*, pages 548–556. International Foundation for Autonomous Agents and Multiagent Systems, 2016.

William. R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3-4):285–294, 1933.

Louis L Thurstone. A law of comparative judgment. *Psychological review*, 34(4):273, 1927.

Tanguy Urvoy, Fabrice Clerot, Raphael Féraud, and Sami Naamane. Generic exploration and k-armed voting bandits. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 91–99, 2013.

Michal Valko, Remi Munos, Branislav Kveton, and Tomas Kocak. Spectral bandits for smooth graph functions. In *Proceedings of the 31st International Conference on Machine Learning*, pages 46–54, 2014.

Fabian Wauthier, Michael Jordan, and Nebojsa Jojic. Efficient ranking from pairwise comparisons. In *International Conference on Machine Learning*, pages 109–117, 2013.

Zheng Wen, Branislav Kveton, and Azin Ashkan. Efficient learning in large-scale combinatorial semi-bandits. In *Proceedings of the 32nd International Conference on Machine Learning*, 2015.

Adrienne Wood, Jared Martin, and Paula Niedenthal. Towards a social functional account of laughter: Acoustic features convey reward, affiliation, and dominance. *PloS one*, 12(8):e0183811, 2017.

Yifan Wu, Roshan Shariff, Tor Lattimore, and Csaba Szepesvári. Conservative bandits. In *International Conference on Machine Learning*, pages 1254–1262, 2016.

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

Yisong Yue and Thorsten Joachims. Beat the mean bandit. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 241–248, 2011.

Yisong Yue, Josef Broder, Robert Kleinberg, and Thorsten Joachims. The k-armed dueling bandits problem. *Journal of Computer and System Sciences*, 78(5):1538–1556, 2012.

Xiaoxue Zhao, Weinan Zhang, and Jun Wang. Interactive collaborative filtering. In *Proceedings of the 22nd ACM International Conference on Information and Knowledge Management*, pages 1411–1420, 2013.

Masrour Zoghi, Shimon Whiteson, Remi Munos, and Maarten de Rijke. Relative upper confidence bound for the k-armed dueling bandit problem. *arXiv preprint arXiv:1312.3393*, 2013.

Masrour Zoghi, Tomas Tunys, Mohammad Ghavamzadeh, Branislav Kveton, Csaba Szepesvari, and Zheng Wen. Online learning to rank in stochastic click models. In *International Conference on Machine Learning*, pages 4199–4208, 2017.

Shi Zong, Hao Ni, Kenny Sung, Nan Rosemary Ke, Zheng Wen, and Branislav Kveton. Cascading bandits for large-scale recommendation problems. In *Proceedings of the 32nd Conference on Uncertainty in Artificial Intelligence*, 2016.