Learning to Rank: Regret Lower Bound and Efficient Algorithms

Richard Combes, Stefan Magureanu, Alexandre Proutière, Cyrille Laroche

Learning to Rank: Regret Lower Bounds and Efficient Algorithms

Richard Combes

Centrale-Supelec,L2S
Gif-sur-Yvette, France
richard.combes@supelec.fr

Stefan Magureanu, Alexandre Proutière, Cyrille Laroche
KTH, Royal Institute of Technology
Stockholm, Sweden
{magur,alepro,laroche}@kth.se

ABSTRACT

Algorithms for learning to rank Web documents, display ads, or other types of items constitute a fundamental component of search engines and more generally of online services. In such systems, when a user makes a request or visits a web page, an ordered list of items (e.g. documents or ads) is displayed; the user scans this list in order, and clicks on the first relevant item if any. When the user clicks on an item, the reward collected by the system typically decreases with the position of the item in the displayed list. The main challenge in the design of sequential list selection algorithms stems from the fact that the probabilities with which the user clicks on the various items are unknown and need to be learned. We formulate the design of such algorithms as a stochastic bandit optimization problem. This problem differs from the classical bandit framework: (1) the type of feedback received by the system depends on the actual relevance of the various items in the displayed list (if the user clicks on the last item, we know that none of the previous items in the list are relevant); (2) there are inherent correlations between the average relevance of the items (e.g. the user may be interested in a specific topic only). We assume that items are categorized according to their topic and that users are clustered, so that users of the same cluster are interested in the same topic. We investigate several scenarios depending on the available side-information on the user before selecting the displayed list: (a) we first treat the case where the topic the user is interested in is known when she places a request; (b) we then study the case where the user cluster is known but the mapping between user clusters and topics is unknown. For both scenarios, we derive regret lower bounds and devise algorithms that approach these fundamental limits.

Categories and Subject Descriptors

I.2.6 [**Computing Methodologies**]: Learning; G.3 [**Mathematics of Computing**]: Probability and Statistics

Keywords

search engines; ad-display optimization; multi-armed bandits; learning

SIGMETRICS'15 June 15 - 19, 2015, Portland, OR, USA

ACM 978-1-4503-3486-0/15/06 ...$15.00.
http://dx.doi.org/10.1145/2745844.2745852.

1. INTRODUCTION

In this paper, we address the problem of learning to rank a set of items based on user feedback. Specifically, we consider a service, where users repeatedly issue queries (e.g. a text string). There are N items, and given the query, a decision maker picks an ordered subset or list of items of size L to be presented to the user. The user examines the items of the list in order, and clicks on the first item she is interested in. The goal for the decision maker is to maximize the number of clicks (over a fixed time horizon, i.e., for a fixed number of queries), and to present the most relevant items in the first slots or positions in the list. The probability for a user to click on an item is unknown to the decision maker initially, and must be learned in an on-line manner, through trial and error. The problem of learning to rank is fundamental in the design of several online services such as search engines [1], ad-display systems [2] and video-on-demand services where the presented items are webpages, ads, and movies, respectively.

The main challenge in the design of learning-to-rank algorithms stems from the prohibitively high number of possible decisions: there are $N!/(N-L)!$ possible lists and typically, we may have more than 1000 items and 10 slots. Hence even trying each decision once can be too costly and inefficient. Fortunately, when selecting a list of items, the decision maker may leverage useful side-information about both the user and her query. For instance, in search engines, the decision maker may be aware of her gender, age, location, etc., and could also infer from her query the type of documents she is interested in (i.e., the topic of her query), which in turn may significantly prune the set of items to choose from. Formally, we assume that the set of items can be categorized into K different disjoint groups, each group corresponding to a given *topic*. Similarly, users are clustered into K classes, such that a class-k user is interested in items in group $h(k)$ only (h is a 1-to-1 mapping from the user classes to the groups of items). This structure simplifies the problem. Two main issues remain however. 1) Even though we could know the class k of the user issuing the query as well as the group of items of interest $h(k)$, we still need to select in that group the L most relevant items. 2) The topic of the query could remain unclear. For example, the query "jaguar" in a search engine may correspond to several topics, for instance a car manufacturer, an animal or a petascale supercomputer. In this case, it seems appropriate to select items from several groups to make sure that at least one item in the list is relevant. This feature is referred to as *diversity principle* in the literature [3]. Another important and final feature of the problem stems from the nature of the decisions: The reward, e.g. the probability that there exists a relevant item in the displayed list, typically exhibits *diminishing returns*, e.g., it can be a submodular function of the set of displayed items [4].

We propose a model that captures both the diversity principle and the diminishing return property, and formalize the problem of designing online learning-to-rank algorithms as a stochastic structured Multi-Armed Bandit (MAB) problem. Stochastic MAB problems [5,6] constitute the most fundamental sequential decision problems with an exploration vs. exploitation trade-off. In such problems, the decision maker selects an arm (here a list of items) in each round, and observes a realization of the corresponding unknown reward distribution. Each decision is based on past decisions and observed rewards. The objective is to maximize the expected cumulative reward over some time horizon by balancing exploitation (arms with higher observed rewards should be selected often) and exploration (all arms should be explored to learn their average rewards). Equivalently, the performance of a decision rule or algorithm can be measured through its expected *regret*, defined as the gap between the expected reward achieved by the algorithm and that achieved by an Oracle algorithm always selecting the best arm. Our MAB problem differs from the classical bandit framework in several ways. First, the type of feedback received by the system depends on the actual relevance of the various items in the displayed list. For example, if the user clicks on the last item, we know that none of the previous items in the list are relevant. Conversely, if the user clicks on the first item, we do not get any feedback for the subsequent items in the list. Then, the rewards of two lists containing a common item are not independent.

There has recently been an important effort to tackle structured MAB problems similar to ours, refer to Section 2 for a survey of existing results. The design of previously proposed learning-to-rank algorithms has been based on heuristics, and these algorithms seem like reasonable solutions to the problem. In contrast, here, our aim is to devise algorithms with provably minimum regret. Our contributions are as follows:

(i) We first investigate the case where the topic of the user query is known. We derive problem-specific regret lower bounds satisfied by *any* algorithm. We also propose PIE (Parsimonious Item Exploration), an algorithm whose regret matches our lower bound, and scales as $O(N_{h(k)}\log(T))$ when applied to queries of class-k users. Here N_k denotes the number of items in the group $h(k)$, and T denotes the time horizon, i.e., the number of queries. The exploration of apparently suboptimal items under PIE is *parsimonious*, as these items are explored only in a single position of the list.

(ii) We then handle the case where the class of the user issuing the query is known, but the group of items she is interested in is not (i.e., the mapping h is unknown). For this scenario, we propose PIE-C (where "C" stands for "Clustered"), an algorithm that efficiently learns the mapping h, and in turn, exhibits the same regret guarantees as PIE, i.e., as if the mapping h was known initially. In fact, we establish that learning the topic of interest for each user class incurs a constant regret (i.e., that does not scale with the time horizon T).

(iii) Finally, we illustrate the performance of PIE and PIE-C using numerical experiments. To this aim, we use both artificially generated data and real-world data extracted from the MovieLens dataset. In all cases, our algorithms outperform existing algorithms.

2. RELATED WORK

Learning to rank relevant contents has attracted a lot of attention in recent years with an increasing trend of modeling the problem as a MAB with semi-bandit feedback. Most of existing models for search engines do not introduce a sufficiently strong structure to allow for the design of efficient algorithms. For example, in [3,4,7–9], the authors hardly impose any structure in their model. Indeed, they consider scenarios where the random variables representing the relevance of the various items are arbitrarily correlated, and even sometimes depart from the stochastic setting by considering adversarial item relevances. The only important structure that these work consider relates to the diminishing return property, and they typically assume that the reward is just a submodular function of the subset of displayed items, see e.g. [4]. As a consequence, the regret guarantees that can be achieved in the corresponding MAB problems (e.g. submodular bandit problems) are weak; a regret with sublinear scaling in the time horizon cannot be achieved. For instance, in submodular bandits, and its variants, the regret has to be defined by considering, as a benchmark, the performance of the best offline polynomial-time algorithm whose approximation ratio is $1-1/e$ [10] unless $NP \subset DTIME(n^{\log\log(n)})$, which indeed implies that the true regret scales linearly with time. In absence of strong structure, one cannot hope to develop algorithms that learn to rank items in a reasonable time. We believe that our model by its additional and natural clustered structure is more appropriate, and in turn, allows us to devise efficient algorithms, i.e., algorithms whose regret scales as $\log(T)$ as the time horizon T grows large.

In [11], Kholi et al. present an analysis close to ours. There, each user is represented by a binary vector in $\{0,1\}^N$ indicating the relevance of the various items to that user, and users are assumed to arrive according to an i.i.d. process with unknown distribution D. They first assume that the relevances of the different items are independent, similar to our setting, and propose a UCB1-based algorithm whose regret provably scales as $O(NL\log(T))$. UCB1 is unfortunately suboptimal, and as we show in this paper, one may devise algorithms with regret scaling as $O(N\log(T))$ in this setting. Then, to extend their results to more general distributions D (allowing for arbitrary correlations among items), the authors of [11] leverage a recent and elegant result from [12] to establish that a regret guarantee scaling as $(1-1/e)T$.

In [13], Slivkins et al. investigate a scenario where items are represented by vectors in a metric space, and assume that their relevance probabilities are Lipschitz continuous. While this model captures the positive correlation between similar items, it does not account for negative correlations between topics. For example, if a user issues the query "jaguar", and if she is not interested in cars, it means that most likely her query concerns the animal.

There has been over the last decade an important research effort towards the understanding of structured MAB problems, see [14] for a recent survey. By structure, we mean that the reward as a function of the arm has some specific properties. Various structures have been investigated in the literature, e.g., Lipschitz [15–18], linear [19], convex [20]. The structure of the MAB problem corresponding to the design of learning-to-rank algorithms is different, and to our knowledge, this paper proposes the first solution (regret lower bounds, and asymptotically optimal algorithms) to this problem.

3. SYSTEM MODEL

3.1 Users, Items, and Side-information

Our model captures the two important properties of online services mentioned in the introduction, namely the diversity principle and the diminishing return property. Let $\mathcal{N}=\{1,..,N\}$ be a set of items (news, articles, files, etc.). Time proceeds in rounds. In each round, a user makes a query and in response to this query, the decision maker has to select from $\mathcal{N}$ an ordered list of L items. We denote by $\mathcal{U}=\{u\subset\mathcal{N}: u=\{u_1,..,u_L\}, u_i\in\mathcal{N}, u_i\neq u_j \text{ if } i\neq j\}$ the set of all possible decisions. The user scans the selected list in order, and stops as soon as she identifies a relevant item. In round n, the relevance of items to the user is captured by

a random vector $X(n) = (X_i(n), i \in \mathcal{N}) \in \{0,1\}^N$, where for any item i, $X_i(n) = 1$ if and only if it is relevant.

Item / User classification. We assume that the set $\mathcal{N}$ is partioned into K disjoint groups $\mathcal{N}_1, \ldots, \mathcal{N}_K$ of respective cardinalities $N_1, \ldots, N_K$. For example, in the case of a query "jaguar", we could consider three groups, corresponding to items related to the animal, the car brand, or a super-computer. This partition of the various items corresponds to the possible broad *topics* of user queries. Similarly, we categorize users into K different classes, and denote by $h(k)$ the index of the topic of interest for class-k users, i.e., the query of class-k users concern items in $\mathcal{N}_{h(k)}$. The mapping h could be known or not as discussed below. Denote by $k(n)$ the class of the user making the query in round n. $(k(n), n \geq 1)$ are i.i.d. random variables with distribution $\phi = (\phi_1, \ldots, \phi_K)$ where $\phi_k = \mathbb{P}[k(n) = k] > 0$. Now, given $k(n) = k$, $(X_i(n), i \in \mathcal{N})$ are independent. Let $\theta_{ki} = \mathbb{P}[X_i(n) = 1 | k(n) = k]$ denote the probability that item i is relevant to class-k users. As already noticed in [9], the above independence assumption captures the diminishing return property. Indeed, given $k(n) = k$, if u is the set of displayed items, the probability that the user finds at least one relevant item in u is $1 - \prod_{i=1}^{L}(1 - \theta_{ku_i})$, which is a submodular function of u (hence with diminishing return).

Observe that the set of users is not specified in our model. We assume that there is an infinite pool of users, that the class of the user issuing a query in round n is drawn from distribution ϕ, and that this user does not issue any query in subsequent rounds. In particular, we cannot learn the class of users from observations. This contrasts with the model proposed in [21], where the set of users is finite, and hence the decision maker can learn the classes of the various users when they repeatedly place queries.

Diversity principle. To capture the diversity principle in our model, we assume that when a user makes a query, she is interested in a single topic only, i.e., in items within a single group $\mathcal{N}_{h(k)}$ only. More precisely, we assume that for all $k, \ell \in [K] := \{1, \ldots, K\}$:

$$\max_{i \in \mathcal{N}_\ell} \theta_{ki} \begin{cases} < \delta & \text{if } \ell \neq h(k), \\ > \Delta & \text{if } \ell = h(k), \end{cases} \tag{1}$$

for some fixed $0 < \delta < \Delta < 1$. Typically, we assume that there is an item that is highly relevant to users of a given class, so that e.g., $\Delta > 1/2$. When in round n, the topic $h(k(n))$ is not known, items of various types should be explored and displayed in the L slots so that the chance of displaying a relevant item is maximized. In other words, (1) captures the diversity principle.

Side-information and Feedback. In round n, under decision rule π, an ordered list of L items is displayed. This decision depends on past observations and some side information, i.e., in round n, the decision rule π maps $((u^\pi(s), f(s), i(s), s < n), i(n))$ to a decision in $\mathcal{U}$, where $u^\pi(s)$, $f(s)$, and $i(s)$ denote the list selected under π, the received feedback, and the side information in round s, respectively.

Feedback: In round n, if the ordered list $u = (u_1, \ldots, u_L)$ is displayed, the decision maker is informed about the first relevant item in the list, i.e., $f(n) = \min\{i \leq L : X_{u_i}(n) = 1\}$. By convention, $f(n) = 0$ if none of the displayed items is relevant. This type of feedback is often referred to as *semi-bandit* feedback in the bandit literature.

Side-information: we model different types of systems depending on the information available to the decision maker about the user placing the query. For example, when a user issues a query, one could infer from her age, gender, location, and other attributes the topic of her query. In such a case, the decision maker knows, before displaying the list of items, the topic of the query, i.e., in round n, $i(n) = h(k(n))$. Alternatively, the decision maker could know the user class (which could be extracted from users' interactions in a social network) but not the topic of her query, i.e., $i(n) = k(n)$. In this case, the mapping h remains unknown.

3.2 Rewards and Regret

To formulate our objectives, we specify the reward of the system when presenting a given ordered list, and introduce the notion of regret which we will aim at minimizing.

The reward is assumed to be a decreasing function of the position of the first relevant of its items, e.g., in search engines, it is preferable to display the most relevant item first. The reward function is denoted by $r(\cdot)$, i.e., the reward is $r(\ell)$ where ℓ denotes the position of the first relevant item in the list. In absence of relevant item, no reward is collected. Without loss of generality, we assume that rewards are in $[0, 1]$.

In view of our assumptions, the expected reward when presenting an ordered list u to a class-k user is:

$$\mu_\theta(u, k) := \sum_{l=1}^{L} r(l)\theta_{ku_l} \prod_{i=1}^{l-1}(1 - \theta_{ku_i}),$$

where $\theta := (\theta_{ki}, k \in [K], i \in \mathcal{N})$ captures the statistical properties of the system.

The performance of a decision rule π is characterised through the notion of regret which compares the performance of an Oracle algorithm aware of the parameter θ to that of the decision rule π up to a given time horizon T (in rounds). The way regret is defined depends on the available side-information. To simplify the presentation and the results, we make the two following assumptions:

(A1) In each group of items, there are at least L items that are relevant with a probability greater than δ. In particular, $N_k \geq L$ for all $k \in [K]$.

(A2) The number of groups K is larger than the number of slots L.

Under these two assumptions, the performance of an Oracle algorithm can be expressed in a simple way. It depends however on the available side-information.

Known topic: When in each round n, the topic $h(k(n))$ is known, the best decision in this round consists in displaying the L most relevant items of group $\mathcal{N}_{h(k(n))}$. For any user class k, and $\ell = h(k)$, for all $i \in [N]$, i_ℓ denotes the item in $\mathcal{N}_\ell$ with the i-th highest relevance: $\theta_{k1_\ell} \geq \theta_{k2_\ell} \geq \ldots \geq \theta_{kN_\ell}$. The list maximizing the expected reward given that the user class is $k(n) = k$ is $u^{\star,k} := (1_{h(k)}, \ldots, L_{h(k)})$. Thus, the expected reward under the Oracle algorithm is:

$$\mu^\star_{1,\theta} := \sum_{k \in [K]} \phi_k \mu_\theta(u^{\star,k}, k),$$

and the regret under algorithm π up to round T is defined as:

$$R^\pi_\theta(T) := T\mu^\star_{1,\theta} - \mathbb{E}\left[\sum_{n=1}^{T} \mu_\theta(u^\pi(n), k(n))\right].$$

To simplify the presentation, we assume that given a user class k, the optimal list is unique, i.e., for any $u \neq u^{\star,k}$, $\mu_\theta(u, k) < \mu_\theta(u^{\star,k}, k)$.

Known user class, and unknown topic: In this case, since the Oracle algorithm is aware of the parameter θ, it is also aware of the mapping h. Thus, the regret of algorithm π is the same as in the previous case, i.e., up to time T, the regret is $R^\pi_\theta(T)$.

Our objective is to devise efficient sequential list selection algorithms in both scenarios, when the topic of successive queries are known, and when only the class of the user issuing the query is known.

4. A SINGLE GROUP OF ITEMS AND USERS

In this section, we study the case where $K = 1$, i.e., there is a single class of user and a single group of item. Even with $K = 1$, our bandit problem remains challenging, due the non-linear reward structure and the reward-specific feedback. To design efficient algorithms, we need to determine *how many* and *where* apparently sub-optimal items should be included for exploration in the displayed list.

When $K = 1$, we can drop the indexes k and $h(k)$. To simplify the notation, we replace $\theta_{ki_{h(k)}}$ by θ_i for all $i \in [N]$. More precisely, we have N items, and users are statistically identical, i.e., θ_i denotes the probability that item i is relevant. Let $\theta = (\theta_1, \ldots, \theta_N)$ and w.l.o.g. the items are ordered so that $\theta_1 \geq \theta_2 \geq \ldots \geq \theta_N$. We denote by $u^\star = (1, \ldots, L)$ the list with maximum expected reward, $\mu^\star_\theta$. The regret of policy π up to round T is then $R^\pi_\theta(T) = T\mu^\star_\theta - \mathbb{E}[\sum_{n=1}^T \mu_\theta(u^\pi(n))]$, where $\mu_\theta(u)$ is the expected reward of list u.

4.1 Regret Lower Bound

We first derive a generic regret lower bound valid for any decreasing reward function $r(\cdot)$. This lower bound will be made more explicit for particular choices of reward functions. We define *uniformly good* algorithms as in [22]. A uniformly good algorithm π satisfies $R^\pi_\theta(T) = o(T^a)$ for all parameters θ and all $a > 0$. Later, it will become clear that such algorithms exist, and therefore we only restrict our attention to the set of such algorithms. We denote by $I(a, b)$ the Kullback-Leibler divergence between two Bernoulli distributions of respective means a and b, i.e., $I(a, b) = a\log(a/b) + (1 - a)\log((1 - a)/(1 - b))$. We further define $\mathcal{U}(i) = \{u \in \mathcal{U} : i \in u\}$, the set of lists in $\mathcal{U}$ that include the item with the i-th highest relevance. Finally, for any list u, and any item $i \in u$, we denote by $p_i(u)$ the position of i in u.

THEOREM 1. *For any uniformly good algorithm π, we have:*

$$\liminf_{T\to\infty} \frac{R^\pi(T)}{\log(T)} \geq c(\theta), \tag{2}$$

where $c(\theta)$ is the minimal value of the objective function in the following optimization problem (P_θ):

$$\inf_{c_u \geq 0, u\in\mathcal{U}} \sum_{u\in\mathcal{U}} c_u(\mu^\star_\theta - \mu_\theta(u)) \tag{3}$$
$$s.t. \sum_{u\in\mathcal{U}(i)} c_u I(\theta_i, \theta_L) \prod_{s<p_i(u)} (1 - \theta_{u_s}) \geq 1, \forall i > L.$$

The solution of the optimization problem (P_θ) has a natural interpretation. For any $u \in \mathcal{U}$, c_u represents the expected number of times the list u should be displayed using an algorithm minimizing the regret. More precisely, u should be displayed $c_u \log(T)$ times asymptotically when the time horizon T grows large. Theorem 1 and its above interpretation are applications of the theory of controlled Markov chains with unknown transition kernel developed in [23]. Next we specify the solution of (P_θ) for two particular classes of reward functions. Define for $i < L$, $\Delta_i = r(i)-r(i+1)$, and $\Delta_L = r(L)$.

1) Reward functions such that: $\Delta_i \geq \Delta_L > 0$ for all $i < L$. This assumption on the reward function seems natural in the context of search engines where the rewards obtained from items presented first are high and rapidly decrease as the position of the item increases.

PROPOSITION 1. *Assume that $\Delta_i \geq \Delta_L > 0$ for $i < L$. Then for all $u \in \mathcal{U}$ such that $u \neq u^\star$, the coefficient c_u corresponding to the solution of (P_θ) satisfies: If for some $i > L$, $u = (1, \ldots, L-1, i)$,*

$$c_u = \frac{1}{I(\theta_i, \theta_L)\prod_{j<L}(1 - \theta_j)},$$

else $c_u = 0$. Hence, we have:

$$c(\theta) = \Delta_L \sum_{i=L+1}^N \frac{\theta_L - \theta_i}{I(\theta_i, \theta_L)}.$$

The above proposition states that very few lists from $\mathcal{U}_k$ should be explored $\Theta(\log(T))$ times. These lists include the $(L-1)$ most relevant items in the $(L-1)$ first slots, and an item that is not within the L most relevant items in the last slot. In other words, an optimal algorithm should include only one sub-optimal item in the list when it explores, and this item should be placed last. This observation will simplify the design of asymptotically optimal algorithms – although of course, initially, the decision maker does not know the $(L-1)$ most relevant items. Note that the minimum regret scales as $(N - L)\log(T)$; this indicates that optimal algorithms should really exploit the reward and feedback structures.

2) Reward functions such that: $\Delta_i = 0$ for all $i < L$, and $\Delta_L > 0$. This scenario may be appropriate in the case of display ads, where the reward obtained when a user clicks does not depend on the position of the ad on the webpage.

PROPOSITION 2. *Assume that $\Delta_i = 0$ for all $i < L$, and $\Delta_L > 0$. Then for all $u \in \mathcal{U}$ such that $u \neq u^\star$, the coefficient c_u corresponding to the solution of (P_θ) satisfies:*
If for some $i > L$, $u = (i, 1, \ldots, L-1)$,

$$c_u = \frac{1}{I(\theta_i, \theta_L)},$$

Else $c_u = 0$. Hence, we have:

$$c(\theta) = \Delta_L \prod_{j<L}(1 - \theta_j) \sum_{i=L+1}^N \frac{(\theta_L - \theta_i)}{I(\theta_i, \theta_L)}.$$

Again the above proposition states that very few lists should be explored $\Theta(\log(T))$ times. These lists are those containing the $(L-1)$ most relevant items in the $(L-1)$ last positions, and an item that is not within the L most relevant items in the first position. In other words, the exploration of items is performed in the first position. Observe that as in the previous case, the minimum regret scales as $(N - L)\log(T)$.

3) General Reward Function. An explicit expression for the lower bound $c(\theta)$ for general reward function is more challenging to derive. However, we suspect that the lists u such that $c_u > 0$ are $w^i_l = (1, \ldots, (l-1), i, l, \ldots, (L-1))$ for some $i > L$. In other words, only one suboptimal item, (i.e., for $i > L$, the i-th most relevant item) is explored at a time, and we should explore i in the l-th position. To determine this position, we make the following heuristic reasoning. Let us fix the number of times i is explored. Given this fixed exploration rate, we select position l that induces the smallest regret. Let us assume that i is explored in slot l. When the list w^i_l is displayed, the probability p_l that i is actually explored is: $p_l = \prod_{j=1}^{l-1}(1 - \theta_j)$. The average number of times i is actually explored when placed in position l is proportional to $1/p_l$. Hence the position $\text{oes}(i)$ where i should be placed should satisfy:

$$\text{oes}(i) \in \operatorname*{argmin}_{l\leq L} f(\theta, w^i_l), \tag{4}$$

where $f(\theta, w_l^i) = \frac{\mu_\theta^\star - \mu_\theta(w_l^i)}{p_l}$. Let $w^i = w^i_{\text{oes}(i)}$. If the argmin in (4) is realized for different positions, we break ties arbitrarily. We state the following conjecture.

For any decreasing reward function $r(\cdot)$, and for $u \in \mathcal{U}$, the coefficient c_u in the solution of (P_θ) has the following form:

$$c_u = \begin{cases} \frac{1}{I(\theta_i, \theta_L) p_{\text{oes}(i)}} & \text{if } u = w^i \text{ for some } i > L, \\ 0 & \text{otherwise.} \end{cases}$$

Observe that the conjecture holds in Cases 1) and 2). In the former, it is optimal to place any suboptimal item i ($i > L$) in the last slot, in which case $p_{\text{oes}(i)} = \prod_{j=1}^{L-1}(1-\theta_j)$. In the latter, it is optimal to place any suboptimal item i in the first slot, in which case $p_{\text{oes}(i)} = 1$.

4.2 Optimal Algorithms

Next we present asymptotically optimal sequential list selection algorithms, i.e., their limiting regret (as T grows large) matches the lower bound derived above. To describe our algorithms, we need to introduce the following definitions. Let $u(n)$ be the list selected in round n, and let $p_i(n)$ denote the position at which item i is shown if $i \in u(n)$, and $p_i(n) = 0$ otherwise. Recall that a sample of θ_i is obtained if and only if $i \in u(n)$ and $X_{i'}(n) = 0$ for all $i' \in \{u_1(n), ..., u_{p_i(n)-1}\}$. Define

$$o_i(n) := \mathbf{1}\left\{i \in u(n), \forall l' < p_i(n), X_{u_{l'}(n)}(n) = 0\right\}.$$

Then we get a sample from θ_i in round n iff $o_i(n) = 1$. Let $t_i(n) := \sum_{n' \le n} o_i(n')$ be the number of samples obtained for θ_i up to round n. The corresponding empirical mean is:

$$\hat{\theta}_i(n) = \frac{1}{t_i(n)} \sum_{n' \le n} o_i(n') X_i(n')$$

if $t_i(n) > 0$ and $\hat{\theta}_i(n) = 0$ otherwise. We also define $c_i(n)$ as the number of times that a list containing i has been selected up to round n: $c_i(n) := \sum_{n' \le n} \mathbf{1}\{i \in u(n')\}$. Let $j(n) = (j_1(n), ..., j_N(n))$ be the indices of the items with empirical means sorted in decreasing order, so that:

$$\hat{\theta}_{j_1(n)}(n) \ge \hat{\theta}_{j_2(n)}(n) \ge ... \ge \hat{\theta}_{j_N(n)}(n)$$

We assume that ties are broken arbitrarily. Define the list of L "leaders" at time n as $\mathcal{L}(n) = (j_1(n), ..., j_L(n))$. The algorithms we propose use the indexes used by the KL-UCB algorithm, known to be optimal in classical MAB problems [24]. The KL-UCB index $b_i(n)$ of item i in round n is:

$$b_i(n) = \max\{q \in [0,1] : t_i(n) I(\hat{\theta}_i(n), q) \le f(n)\},$$

where $f(n) = \log(n) + 4\log(\log(n))$. Let:

$$\mathcal{B}(n) := \{i \notin \mathcal{L}(n) : b_i(n) \ge \hat{\theta}_{j_L(n)}(n)\}.$$

be the set of items which are not in the set of leaders, and whose index are larger than the empirical mean of item $j_L(n)$. Intuitively, $\mathcal{B}(n)$ includes items which are potentially better than the worst current leader. For $1 \le i \le N$, define decision:

$$U_i^l(n) = (j_1(n), ..., j_{l-1}(n), i, j_l(n), ..., j_{L-1}(n)).$$

$U_i^l(n)$ is the list obtained by considering the $L-1$ first items of $\mathcal{L}(n)$, and by placing item i at position l. We are now ready to present our algorithms. The latter, referred to as PIE(l), are parametrized by $l \in \{1, \dots, L\}$, the position where the exploration is performed. In round n, PIE(l) proceeds as follows:

(i) if $\mathcal{B}(n)$ is empty, then the leader is selected: $u(n) = \mathcal{L}(n)$;

(ii) otherwise, we select $u(n) = \mathcal{L}(n)$ with probability $1/2$, and $u(n) = U^l_{i(n)}(n)$ with probability $1/2$, where $i(n)$ is chosen from $\mathcal{B}(n)$ uniformly at random.

Algorithm PIE(l)

Init: $\mathcal{B}(1) = \emptyset$, $\hat{\theta}_i(1) = 0 = b_i(1)\ \forall i$, $\mathcal{L}(1) = \{1, \dots, L\}$
For $n \ge 1$**:**
If $\mathcal{B}(n) = \emptyset$, select $\mathcal{L}(n)$
Else w.p.1/2, select $\mathcal{L}(n)$, w.p. select $U_I^l(n)$, $I \in \mathcal{B}(n)$ unif. distributed
Compute: $\mathcal{B}(n+1)$, $\mathcal{L}(n+1)$, and $\hat{\theta}_i(n+1)$, $b_i(n+1)$, $\forall i$

Refer to pseudo-code of PIE(l) for a formal description. Note that the PIE(l) algorithm has low computational complexity. It can be easily checked that it requires at each round $O(N + L\log(N))$ operations. In the following theorem, we provide a finite-time regret upper bound of the PIE(l) algorithm. Introduce $\eta = \prod_{i=1}^{L-1}(1-\theta_i)^{-1}$ and recall that $p_l = \prod_{i'<l}(1-\theta_{i'})$ and $u^{i,l} = (1, \dots, (l-1), i, (l+1), \dots, (L-1))$ for all $i > l$.

THEOREM 2. *Under algorithm π=PIE(l), for all $T \ge 1$ and, all $\epsilon > 0$ and all $0 < \delta < \delta_0 = \min_{i<N}(\theta_i - \theta_{i+1})/2$, the regret under π satifies:*

$$R^\pi(T) = f(T) c^{PIE(l)}(\theta, \delta) + C(\theta, \delta, \epsilon),$$

where

$$c^{PIE(l)}(\theta, \delta) = p_l^{-1} \sum_{i=L+1}^{N} \frac{\mu(u^\star) - \mu(u^{i,l})}{I(\theta_i + \delta, \theta_L - \delta)},$$

$$C(\theta, \delta, \epsilon) = 2N\eta[(5 + 8NL)\eta + (3 + 2L)\delta^{-2}] + 15L + (N-L)p_l^{-1}\left[\epsilon^{-2}p_l^{-1} + \delta^{-2}(1-\epsilon)^{-1}\right].$$

As a consequence:

$$\limsup_{T\to\infty} \frac{R^\pi(T)}{\log(T)} \le c^{PIE(l)}(\theta) := \sum_{i=L+1}^{N} \frac{\mu(u^\star) - \mu(u^{i,l})}{\prod_{i'<l}(1-\theta_{i'}) I(\theta_i, \theta_L)}.$$

A direct consequence of the above theorem is that PIE(L) and PIE(1) are asymptotically optimal in Case 1) (convex decreasing reward functions) and Case 2) (constant rewards), respectively. Indeed, one can easily check that for example $c^{\text{PIE}(L)}(\theta) = c(\theta)$ in Case 1).

4.3 Proofs: Lower Bounds

4.3.1 Proof of Theorem 1

The result is a consequence of the theory of controlled Markov chain with unknown transition rates [23]. We apply the formalism of [23] as follows. The state space of the Markov chain is $\mathcal{X} = \{0, 1, .., L\}$, and the state will capture the feedback obtained from the previous decision, i.e., $x = 0$ means that no item in the list is relevant, and $x = i$ means that the first relevant item is in position i. The set of control actions is the set of lists $\mathcal{U}$. The transition probability from state x to state y given that the chosen list is u is $p(x, y; u, \theta)$ where

$$p(x, y; u, \theta) = p(y; u, \theta) = \begin{cases} \prod_{s=1}^{L}(1 - \theta_{u_s}) \text{ if } y = 0, \\ \theta_{u_y} \prod_{s=1}^{y-1}(1 - \theta_{u_s}) \text{ if } y \in \{1, .., L\}. \end{cases}$$

The reward associated to the state x and the control action u is denoted by $g(x, u)$, and here we have $g(x, u) = r(x)$. Finally, the set of control laws is $\mathcal{G} = \mathcal{U}$. The expected reward under the control law u is $\mu_\theta(u)$. Next we apply Theorem 1 in [23]. To this aim, we first introduce the KL divergence between two parameters $\lambda \in [0, 1]^N$ and θ under control law u as:

$$I^u(\lambda, \theta) = \sum_{s=1}^{L} \theta_{u_s} \left[\prod_{i=1}^{s-1}(1 - \theta_{u_i})\right] \log\left(\frac{\theta_{u_s} \prod_{i=1}^{s-1}(1 - \theta_{u_i})}{\lambda_{u_s} \prod_{i=1}^{s-1}(1 - \lambda_{u_i})}\right) + \left[\prod_{i=1}^{s}(1 - \theta_{u_i})\right] \log\left(\frac{\prod_{i=1}^{s}(1 - \theta_{u_i})}{\prod_{i=1}^{s}(1 - \lambda_{u_i})}\right),$$

which can be rewritten as:

$$I^u(\theta, \lambda) = \sum_{i=1}^{L} I(\theta_{u_i}, \lambda_{u_i}) \prod_{s=1}^{i-1}(1 - \theta_{u_s}).$$

Let us further introduce the set of *bad* parameters $B(\theta)$ as:

$$B(\theta) = \{\lambda \in [0, 1]^N : I^{u^\star}(\theta, \lambda) = 0 \text{ and } \exists u \neq u^\star, \mu_\lambda(u) > \mu^\star_\theta\},$$

where $\mu_\lambda(u)$ denotes the expected reward of decision u under parameter λ. By definition, if $\lambda \in B(\theta)$, there is $i > L$ such that $\lambda_i > \theta_L$. Thus we can decompose $B(\theta)$ into the union of sets $B_i(\theta) = \{\lambda \in B(\theta), \lambda_i > \theta_L\}$ over $i \in \{L + 1, .., N\}$. By Theorem 1 in [23], we have, for any uniformly good algorithm π:

$$\lim \inf_{T\to\infty} \frac{R^\pi(T)}{\log(T)} \geq c(\theta),$$

where

$$c(\theta) = \inf_{c_u \geq 0, u \in \mathcal{U}} \sum_{u \neq u^\star} c_u(\mu^*_\theta - \mu_\theta(u))$$
$$\text{s. t. } \forall i > L, \inf_{\lambda \in B_i(\theta)} \sum_{u \neq u^\star} c_u I^u(\theta, \lambda) \geq 1.$$

By definition of $B_i(\theta)$, if $\lambda \in B_i(\theta)$, then $\lambda_i > \theta_L$. It can easy seen that $\inf_{\lambda \in B_i(\theta)} I^u(\theta, \lambda)$ is achieved for some parameter $\lambda^\star$ such that $\lambda^\star_i = \theta_L$ and $\lambda^\star_j = \theta_j$ for $j \neq i$ and hence:

$$\inf_{\lambda \in B_i(\theta)} I^u(\theta, \lambda) = \sum_{u \in \mathcal{U}(i)} c_u \prod_{s < p_i(u)} (1 - \theta_{u_s}) I(\theta_i, \theta_L) \geq 1.$$

This completes the proof. □

4.3.2 Proof of Proposition 1

For $i > L$, we define v^i the list such that $v^i_j = j$ for $j < l$, and $v^i_L = i$. According to Proposition 1, these lists only should be explored under an optimal algorithm. Let $c = \{c_u : u \neq u^\star\}$ be a solution of the LP introduced in Theorem 1. We prove by contradiction that $c_u > 0$ implies that there exists $i > l$ such that $u = v^i$. Assume $\exists u \neq u^\star$ such that $c_u > 0$ and $u \neq v^i, \forall i > L$. We propose a new set of coefficients $c' = \{c'_u : u \neq u^\star\}$ such the value of objective function $c'(\theta)$ of the LP under c' is less than under c. We use the following notation: $c_{w,i} = c_v \frac{\prod_{s \leq i}(1-\theta_{w_s})}{\prod_{s<L}(1-\theta_s)}$ for any $w \in \mathcal{U}$. Recall that $p_i(w)$ is the position of i in w. Now introduce c' such that for all $u \neq u^\star$:

$$c'_w = \begin{cases} 0 & \text{if } w = u, \\ c_w + c_{u,p_i(u)} & \text{if } \exists i > L \text{ such that } w = v^i, \\ c_w & \text{otherwise.} \end{cases}$$

We show that c' yields a strictly lower value of the objective function in the LP of Theorem 1 than c, a contradiction. Denote by $c(\theta)$ and $c'(\theta)$ the value of the objective function of the LP under c and c', respectively. We have:

$$c(\theta) - c'(\theta) = c_u(\mu^*_\theta - \mu_\theta(u)) - \sum_{i:u_i>L} c_{u,i}(\mu^*_\theta - \mu_\theta(v^{u_i})).$$

It is easy to check that: $\mu_\theta(u) = r(1) - \sum_{l=1}^{L} \Delta_l \prod_{s \leq l}(1 - \theta_{u_s})$. Therefore $\mu^*_\theta - \mu_\theta(u) = \sum_{i=1}^{L} \Delta_i(\prod_{s \leq i}(1 - \theta_{u_s}) - \prod_{s \leq i}(1 - \theta_s))$. Since $\Delta_L = r(L)$, we have:

$$\mu^*_\theta - \mu_\theta(u) - \sum_{i:u_i>L} \frac{\prod_{s<i}(1 - \theta_{u_j})}{\prod_{s<L}(1 - \theta_j)}(\mu^*_\theta - \mu_\theta(v^{u_i})) = \sum_{i=1}^{L} \Delta_i(\prod_{s \leq i}(1 - \theta_{u_s}) - \prod_{s \leq i}(1 - \theta_s)) - \sum_{i:u_i>L} \Delta_L \prod_{s<i}(1 - \theta_{u_s})(\theta_L - \theta_{u_i}).$$

Let $i \leq L$ such that $u_i > L$. We have:

$$\Delta_i(\prod_{s \leq i}(1 - \theta_{u_s}) - \prod_{s \leq i}(1 - \theta_s)) - \Delta_L \prod_{s<i}(1 - \theta_{u_s})(\theta_L - \theta_{u_i}) \geq \Delta_i(\prod_{s<i}(1 - \theta_{u_s})(1 - \theta_L) - \prod_{s \leq i}(1 - \theta_s)) > 0.$$

We deduce:

$$c_u(\mu^*_\theta - \mu_\theta(u)) > \sum_{\substack{i \\ u_i>L}} c_{u,i}(\mu^*_\theta - \mu_\theta(v^{u_i})).$$

And hence, $c'(\theta) < c(\theta)$. We have shown that in c the solution of the LP involved in Theorem 1, $c_u > 0$ iff $\exists i > l$: $u = v^i$. Now we can easily solve the LP in light of this result, and show that the c_u's are of the form as stated in Proposition 1. The proof of Proposition 2 is similar. □

4.4 Proof: Regret Upper bound for PIE(l)

4.4.1 Preliminaries

Before analyzing the regret of PIE(l), we state and prove Lemma 1. The latter shows that, under algorithm PIE(l), the set of rounds at which either
(i) the set of leaders is different from the optimal decision, or
(ii) the empirical mean of one of the leaders deviates from its expectation by more than a fixed quantity $\delta > 0$,
has finite size (in expectation). Note that (i) and (ii) are not mutually exclusive. The upper bound provided by Lemma 1 is explicit as a function of the parameters $(\theta_i)_i$ and δ.

LEMMA 1. *Define* $\delta_0 = \min_{i<N}(\theta_i - \theta_{i+1})/2$ *and* $\eta = \prod_{i=1}^{L-1}(1-\theta_i)^{-1}$. *Let* $0 < \delta < \delta_0$ *and define the following sets of rounds:*

$$\mathcal{A} = \{n \geq 1 : \mathcal{L}(n) \neq u^\star\},$$
$$\mathcal{D} = \{n \geq 1 : \exists i \in \mathcal{L}(n) : |\hat{\theta}_i(n) - \theta_i| \geq \delta\}.$$

and $\mathcal{C} = \mathcal{A} \cup \mathcal{D}$. *Under algorithm PIE(l), for all* $0 < \delta < \delta_0$ *we have:*

$$\mathbb{E}[|\mathcal{C}|] \leq 2N\eta[(5+8NL)\eta + (3+2L)\delta^{-2}] + 15L.$$

Proof. Fix $\delta < \delta_0$ throughout the proof. Our goal is to upper bound the expected size of $\mathcal{C}$. To do so, we decompose $\mathcal{C}$ in an appropriate manner. We introduce the following sets of instants:

$$\mathcal{E} = \{n \geq 1 : \exists i \in \{1, ..., L\} : b_i(n) \leq \theta_i\}$$
$$\mathcal{G} = \{n \geq 1 : n \in \mathcal{A} \setminus (\mathcal{D} \cup \mathcal{E}), \exists i \in \{1, ..., L\} \setminus \mathcal{L}(n) : |\hat{\theta}_i(n) - \theta_i| \geq \delta\}.$$

We first check that $\mathcal{C} \subset \mathcal{D} \cup \mathcal{E} \cup \mathcal{G}$. Since $\mathcal{C} = \mathcal{A} \cup \mathcal{D}$, it is sufficient to prove that $\mathcal{A} \subset (\mathcal{D} \cup \mathcal{E} \cup \mathcal{G})$. Let $n \in \mathcal{A} \setminus (\mathcal{D} \cup \mathcal{E})$. Let $i, i' \in \mathcal{L}(n)$, with $i < i'$. Since $n \notin \mathcal{D}$ we have $|\hat{\theta}_i(n) - \theta_i| \leq \delta$, $|\hat{\theta}_{i'}(n) - \theta_{i'}| \leq \delta$, and $\delta \leq (\theta_i - \theta_{i'})/2$, therefore $\hat{\theta}_i(n) \geq \hat{\theta}_{i'}(n)$. This proves that $(j_1(n), ..., j_L(n))$ is an increasing sequence. We have that $j_L(n) > L$, otherwise we have $(j_1(n), ..., j_L(n)) = (1, 2, ..., L)$ hence $\mathcal{L}(n) = u^\star$ and $n \notin \mathcal{A}$, a contradiction. Since $j_L(n) > L$ there exists $i \leq L$ such that $i \notin \mathcal{L}(n)$. Let us now prove by contradiction that $|\hat{\theta}_i(n) - \theta_i| \geq \delta$. Assume that $|\hat{\theta}_i(n) - \theta_i| \leq \delta$, then we have $|\hat{\theta}_{j_L(n)}(n) - \theta_{j_L(n)}| \leq \delta$ (since $j_L(n) \in \mathcal{L}(n)$ and $n \notin \mathcal{D}$) so that $\hat{\theta}_i(n) > \hat{\theta}_{j_L(n)}(n)$. In turn this would imply that $i \in \mathcal{L}(n)$ which is a contradiction. Finally we have proven that $n \in \mathcal{A} \setminus (\mathcal{D} \cup \mathcal{E})$ implies $n \in \mathcal{G}$. Hence $\mathcal{C} \subset \mathcal{D} \cup \mathcal{E} \cup \mathcal{G}$, and by a union bound:

$$\mathbb{E}[|\mathcal{C}|] \leq \mathbb{E}[|\mathcal{D}|] + \mathbb{E}[|\mathcal{E}|] + \mathbb{E}[|\mathcal{G}|].$$

Next we prove the following inequalities:
(a) $\mathbb{E}[|\mathcal{D}|] \leq 2N\eta\left[10\eta + 3\delta^{-2}\right]$;
(b) $\mathbb{E}[|\mathcal{E}|] \leq 15L$;
(c) $\mathbb{E}[|\mathcal{G}|] \leq 4NL\eta\left[4\eta + \delta^{-2}\right]$.

Inequality (a): We further decompose $\mathcal{D}$ as $\mathcal{D} = \cup_{i=1}^{N}(\mathcal{D}_{i,1} \cup \mathcal{D}_{i,2})$, with:

$$\mathcal{D}_{i,1} = \{n \geq 1 : i \in \mathcal{L}(n), j_L(n) \neq i, |\hat{\theta}_i(n) - \theta_i| \geq \delta\}$$
$$\mathcal{D}_{i,2} = \{n \geq 1 : i \in \mathcal{L}(n), j_L(n) = i, |\hat{\theta}_i(n) - \theta_i| \geq \delta\}$$

In other words, $\mathcal{D}_{i,1}$ is the set of rounds at which i is not the L-th leader, so that if $n \in \mathcal{D}_{i,1}$ then i will be included in $u(n)$. $\mathcal{D}_{i,2}$ is the set of instants at which i is the L-th leader, so that if $n \in \mathcal{D}_{i,2}$, then either i or $i(n)$ will be included in $u(n)$.

First let $n \in \mathcal{D}_{i,1}$. Then we have $i \in u(n)$ by definition of the algorithm. Hence $\mathbb{E}[o_i(n)|n \in \mathcal{D}_{i,1}] \geq \eta^{-1}$. Furthermore, for all n, $\mathbf{1}\{n \in \mathcal{D}_{i,1}\}$ is $\mathcal{F}_{n-1}$ measurable ($\mathcal{F}_{n-1}$ the σ-algebra generated by $u(s)$ and the corresponding feedback for $s \leq n-1$). Therefore we can apply the second statement of Lemma 5, presented in Appendix (with $H := \mathcal{D}_{i,1}$, $c := \eta^{-1}$) to obtain: $\mathbb{E}[|\mathcal{D}_{i,1}|] \leq 2\eta\left[2\eta + \delta^{-2}\right]$.

Next let $n \in \mathcal{D}_{i,2}$. Then we have that $i \in u(n)$ with probability at least $1/2$ by definition of the algorithm, so that $\mathbb{E}[o_i(n)|n \in \mathcal{D}_{i,2}] \geq \eta^{-1}/2$. Also $\mathbf{1}\{n \in \mathcal{D}_{i,2}\}$ is $\mathcal{F}_{n-1}$ measurable. Hence applying the second statement of Lemma 5 (with $H \equiv \mathcal{D}_{i,2}$, $c \equiv \eta^{-1}/2$) we obtain: $\mathbb{E}[|\mathcal{D}_{i,2}|] \leq 4\eta\left[4\eta + \delta^{-2}\right]$.

Applying a union bound over $1 \leq i \leq N$, we get:

$$\mathbb{E}[|\mathcal{D}|] \leq \sum_{i=1}^{N} \mathbb{E}[|\mathcal{D}_{i,1}|] + \mathbb{E}[|\mathcal{D}_{i,2}|] \leq 2N\eta\left[10\eta + 3\delta^{-2}\right].$$

Inequality (b): Decompose $\mathcal{E}$ as $\mathcal{E} = \cup_{i=1}^{L}\mathcal{E}_i$ where

$$\mathcal{E}_i = \{n \geq 1 : b_i(n) \leq \theta_i\}.$$

Applying Lemma 6 we obtain that $\mathbb{E}[|\mathcal{E}_i|] \leq 15$ for all i, so that:

$$\mathbb{E}[|\mathcal{E}|] \leq \sum_{i=1}^{L} \mathbb{E}[|\mathcal{E}_i|] \leq 15L.$$

Inequality (c): Decompose $\mathcal{G}$ as $\mathcal{G} = \cup_{i=1}^{L}\mathcal{G}_i$ where

$$\mathcal{G}_i = \{n \geq 1 : n \in \mathcal{A} \setminus (\mathcal{D} \cup \mathcal{E}), i \notin \mathcal{L}(n), |\hat{\theta}_i(n) - \theta_i| \geq \delta\}.$$

For a given $i \leq L$, $\mathcal{G}_i$ is the set of rounds at which i is not one of the leaders, and is not accurately estimated. Let $n \in \mathcal{G}_i$. Since $i \notin \mathcal{L}(n)$, we must have $j_L(n) > L$. In turn, since $n \notin \mathcal{D}$ we have $|\hat{\theta}_{j_L(n)}(n) - \theta_{j_L(n)}| \leq \delta$, so that

$$\hat{\theta}_{j_L(n)}(n) \leq \theta_{j_L(n)} + \delta \leq \theta_{L+1} + \delta \leq (\theta_{L+1} + \theta_L)/2.$$

Furthermore, since $n \notin \mathcal{E}$ and $1 \leq i \leq L$, we have $b_i(n) \geq \theta_i \geq \theta_L \geq (\theta_{L+1} + \theta_L)/2 \geq \hat{\theta}_{j_L(n)}(n)$. This implies that $i \in \mathcal{B}(n)$. Since $i(n)$ has uniform distribution over $\mathcal{B}(n)$, we have that $i(n) = i$ with probability at least $1/N$. We have that for all n, $\mathbf{1}\{n \in \mathcal{G}_i\}$ is $\mathcal{F}_{n-1}$ measurable. Further, $\mathbb{E}[o_i(n)|n \in \mathcal{G}_i] \geq \eta^{-1}/(2N)$. So we can apply Lemma 5 (with $H \equiv \mathcal{G}_i$ and $c \equiv \eta^{-1}/(2N)$) to yield: $\mathbb{E}[|\mathcal{G}_i|] \leq 4N\eta\left[4N\eta + \delta^{-2}\right]$

Using a union bound over $1 \leq i \leq L$, we obtain:

$$\mathbb{E}[|\mathcal{G}|] \leq \sum_{i=1}^{L} \mathbb{E}[|\mathcal{G}_i|] \leq 4NL\eta\left[4N\eta + \delta^{-2}\right].$$

Putting inequalities (a), (b) and (c) together, we obtain the announced result:

$$\mathbb{E}[|\mathcal{C}|] \leq \mathbb{E}[|\mathcal{D}|] + \mathbb{E}[|\mathcal{E}|] + \mathbb{E}[|\mathcal{G}|] \leq 2N\eta[(5+8NL)\eta + (3+2L)\delta^{-2}] + 15L,$$

which concludes the proof. □

4.4.2 *Proof of Theorem 2*

We decompose the regret by distinguishing rounds in $\mathcal{C}$ (as defined in the statement of Lemma 1), and other rounds. For all $i > L$, we define the sets of instants between 1 and T at which $n \notin \mathcal{C}$ and decision $u^{i,l}$ is selected (recall that $u^{i,l} = (1, \ldots, (l-1), i, (l+1), \ldots, (L-1))$):

$$\mathcal{K}_i = \{1 \leq n \leq T : n \notin \mathcal{C}, \mathcal{L}(n) = u^\star, u(n) = u^{i,l}\}.$$

By design of the algorithm, when $n \notin \mathcal{C}$, the leader is the optimal decision, and so the only sub-optimal decisions that can be selected are $\{u^{L+1,l}, ..., u^{N,l}\}$. Hence the set of instants at which a suboptimal decision is selected verifies:

$$\{1 \leq n \leq T : u(n) \neq u^\star\} \subset \mathcal{C} \cup (\cup_{i=L+1}^{N}\mathcal{K}_i).$$

Since $\mu(u^\star) - \mu(u) \leq 1$ for all u, we obtain the upper bound:

$$R^\pi(T) \leq \mathbb{E}[|\mathcal{C}|] + \sum_{i=L+1}^{N}\left[\mu(u^\star) - \mu(u^{i,l})\right]\mathbb{E}[|\mathcal{K}_i|].$$

By Lemma 1, we have:

$$\mathbb{E}[|\mathcal{C}|] \leq 2N\eta[(5+8NL)\eta + (3+2L)\delta^{-2}] + 15L.$$

Hence, to complete the proof, it is sufficient to prove that, for all $i \geq L+1$, all $\epsilon > 0$ and all $0 < \delta < \theta_L - \theta_{L+1}$, we have:

$$\mathbb{E}[|\mathcal{K}_i|] \leq p_l^{-1} \frac{f(T)}{(1-\epsilon) I(\theta_i + \delta, \theta_L - \delta)} + p_l^{-1} \left[p_l^{-1} \epsilon^{-2} + \delta^{-2} (1-\epsilon)^{-1} \right]. \quad (5)$$

Define the number of rounds in $\mathcal{K}_i$ before round n: $k_i(n) = \sum_{n' \leq n} \mathbf{1}\{n' \in \mathcal{K}_i\}$. Fix $\epsilon > 0$, define $t_0 = f(T)/I(\theta_i + \delta, \theta_L - \delta)$, and define the following subsets of $\mathcal{K}_i$:

$$\mathcal{K}_{i,1} = \left\{ n \in \mathcal{K}_i : t_i(n) \leq p_l(1-\epsilon) k_i(n) \text{ or } |\hat{\theta}_i(n) - \theta_i| \geq \delta \right\},$$

$$\mathcal{K}_{i,2} = \left\{ n \in \mathcal{K}_i : t_0 \leq p_l(1-\epsilon) k_i(n) \right\}.$$

Namely, $\mathcal{K}_{i,1}$ is the set of rounds in $\mathcal{K}_i$ where either item i has been sampled (we recall that i is sampled iff all items presented before i where not relevant) less than $p_l(1-\epsilon)k_i(n)$ times or and its empirical mean deviates from its expectation by more than δ. $\mathcal{K}_{i,2}$ is the number of instants in $\mathcal{K}_i$ where $p_l(1-\epsilon)k_i(n)$ is smaller than t_0, i.e $\mathcal{K}_{i,2}$ is the set of the first $t_0 p_l^{-1}(1-\epsilon)^{-1}$ instants of $\mathcal{K}_i$.

Let us prove that $\mathcal{K}_i \subset \mathcal{K}_{i,1} \cup \mathcal{K}_{i,2}$. We proceed by contradiction: Consider $n \in \mathcal{K}_i \setminus (\mathcal{K}_{i,1} \cup \mathcal{K}_{i,2})$. We prove that we have both (a) $t_i(n) \geq t_0$ and (b) $b_i(n) \geq \theta_L - \delta$. Since $n \notin \mathcal{K}_{i,1}$ we have that $t_i(n) \geq p^{-1}(1-\epsilon)k_i(n)$ and since $n \notin \mathcal{K}_{i,2}$ we have $p_l(1-\epsilon)k_i(n) \geq t_0$. So (a) holds. By definition of the algorithm, we have that $i \in \mathcal{B}(n)$, so that $b_i(n) \geq \hat{\theta}_{j_L(n)}(n)$. Furthermore, since $n \in \mathcal{K}_i$ we have that $n \notin \mathcal{C}$, so that $j_L(n) = L$, and $|\hat{\theta}_L(n) - \theta_L| \leq \delta$. In turn, this implies $b_i(n) \geq \hat{\theta}_{j_L(n)}(n) = \hat{\theta}_L(n) \geq \theta_L - \delta$ so (b) holds as well. Combining (a) and (b) with the definition of $b_i(n)$:

$$t_0 I(\hat{\theta}_i(n), \theta_L - \delta) \leq t_i(n) I(\hat{\theta}_i(n), \theta_L - \delta) \leq f(n) \leq f(T),$$

and thus: $I(\hat{\theta}_i(n), \theta_L - \delta) \leq I(\theta_i - \delta, \theta_L - \delta)$, which proves that $|\hat{\theta}_i(n) - \theta_i| \geq \delta$ using the fact that the function $x \mapsto I(x,y)$ is decreasing for $0 \leq x \leq y$. Hence $n \in \mathcal{K}_{i,1}$ which is a contradiction since we assumed that $n \in \mathcal{K}_i \setminus (\mathcal{K}_{i,1} \cup \mathcal{K}_{i,2})$. Hence $\mathcal{K}_i \subset \mathcal{K}_{i,1} \cup \mathcal{K}_{i,2}$ as announced. We now provide upper bounds on the expected sizes of $\mathcal{K}_{i,1}$ and $\mathcal{K}_{i,2}$.

Set $\mathcal{K}_{i,1}$: Since $n \in \mathcal{K}_{i,1} \subset \mathcal{K}_i$ implies $u(n) = u^{i,l}$ we have that $\mathbb{E}[o_i(n) | n \in \mathcal{K}_{i,1}] = p_l$. Applying Corollary 1 presented in Appendix (with $H \equiv \mathcal{K}_{i,1}$ and $c \equiv p_l$) we obtain:

$$\mathbb{E}[|\mathcal{K}_{i,1}|] \leq p_l^{-1} \left[p_l^{-1} \epsilon^{-2} + \delta^{-2} (1-\epsilon)^{-1} \right].$$

Set $\mathcal{K}_{i,2}$: Since $n \in \mathcal{K}_{i,2}$ implies that $k_i(n) \leq t_0 p_l^{-1}(1-\epsilon)^{-1}$ and that $k_i(n)$ is incremented at n, we have that:

$$\mathbb{E}[|\mathcal{K}_{i,2}|] \leq t_0 p_l^{-1} (1-\epsilon)^{-1}.$$

Putting it all together we obtain the desired bound (5) on the expected size of $\mathcal{K}_i$, which concludes the proof of the first statement of Theorem 2. The second statement of the theorem is obtained by taking the limit $T \to \infty$ and then $\delta \to 0$. □

5. KNOWN TOPIC

In the remaining of the paper, we consider $K > 1$ groups of users and items, and switch back to the notations introduced in Section 3. In this section, we consider the scenario where in each round n, the topic of the request is known, i.e., the decision maker is informed about $h(k(n))$ before selecting the items to be displayed. In such a scenario, the problem of the design of sequential list selection algorithms can be decomposed into K independent bandit problems, one for each topic. Indeed in view of Assumption (A1), when the topic of the request is $h(k)$, any algorithm should present, in the list, items from $\mathcal{N}_{h(k)}$ only. The K independent MAB problems are instances of the problems considered in the previous section. As a consequence, we can apply the analysis of Section 4, and immediately deduce regret lower bounds and asymptotically optimal algorithms. Optimal algorithms are obtained by just running K independent PIE(l) algorithms, one for each topic. We refer to as $K\times$PIE(l) the resulting global algorithm.

Define $\mathcal{U}_k$ as the set of lists containing items from $\mathcal{N}_{h(k)}$ only, i.e., $\mathcal{U}_k := \{u \in \mathcal{U} : \forall s \in [L], u_s \in \mathcal{N}_{h(k)}\}$. We denote by $\mathcal{U}_k(i) = \{u \in \mathcal{U}_k : i_{h(k)} \in u\}$, the set of lists in $\mathcal{U}_k$ that include the item $i_{h(k)}$ with the i-th highest relevance in $\mathcal{N}_{h(k)}$. Finally, for $u \in \mathcal{U}_k(i)$, we refer to as $p_i(u)$ as the position of $i_{h(k)}$ in the list u. The following theorem is a direct consequence of Theorem 1.

THEOREM 3. *Let $\theta \in [0,1]^{K \times N}$. For any uniformly good algorithm π, we have:*

$$\liminf_{T \to \infty} \frac{R^\pi(T)}{\log(T)} \geq \sum_{k \in [K]} c_k(\theta), \quad (6)$$

where for any $k \in [K]$, $c_k(\theta)$ is the minimal value of the objective function in the following optimization problem $(P_{\theta,k})$:

$$\inf_{c_u \geq 0, u \in \mathcal{U}_k} \sum_{u \in \mathcal{U}_k} c_u (\mu_\theta^{\star,k} - \mu_\theta(u,k)) \quad (7)$$

$$\text{s.t.} \sum_{u \in \mathcal{U}_k(i)} c_u \prod_{s < p_i(u)} (1 - \theta_{k u_s}) I(\theta_{k i_{h(k)}}, \theta_{k L_{h(k)}}) \geq 1, \quad \forall i > L.$$

The LPs $(P_{\theta,k})$ are similar to (P_θ) presented in Theorem 1, and enjoy the same simplifications (see Propositions 1 and 2) when the reward function has the specific structure of Case 1) or 2). Observe that the regret lower bound does not depend on the proportions of queries made by users of the various classes (remember that we assumed that $\phi_k > 0$ for all $k \in [K]$) – this is simply due to the facts that over the time horizon T, we roughly have $\phi_k T$ queries generated by class-k users, and that the regret incurred for class-k users is $c_k(\theta)\log(\phi_k T) \approx c_k(\theta)\log(T)$.

The next theorem is a direct consequence of Theorem 2, and states that $K\times$PIE(L) and $K\times$PIE(1) are asymptotically optimal in Cases 1) and 2), respectively.

THEOREM 4. *Assume that the reward function has the specific structure described in Case 1) (resp. 2)). Under algorithm $\pi = K\times$PIE(L) (resp. $\pi = K\times$PIE(1)), we have for all θ:*

$$\limsup_{T \to \infty} \frac{R^\pi(T)}{\log(T)} \leq \sum_{k \in [K]} c_k(\theta).$$

6. KNOWN USER-CLASS AND UNKNOWN TOPIC

In this section, we address the problem with $K > 1$ groups of users and items, and where in each round n, the decision maker is aware of the class of the user issuing the query, but does not know the mapping h, i.e., initially, the decision maker does not know which topic the users of the various classes are interested in. Of course, this scenario is more challenging than the one where, before selecting a list of items, the decision maker is informed on the topic $h(k(n))$, and hence, the regret lower bound described in Theorem 3 is still valid.

Next we devise a sequential list selection algorithm that learns the mapping h very rapidly. More precisely, we prove that its asymptotic regret satisfies the same regret upper bound as those

Algorithm PIE-C(l, d)

Init: $\hat{\theta}_{ki}(1) = 0, \forall i, k$
For $n \geq 1$**:**
Get the class $k(n)$ of the user issuing the query, and compute $C(n) = \{h \in [K] : \max_{i \in \mathcal{N}_h} \hat{\theta}_{k(n)i}(n) \geq d\}$
If $C(n) = \emptyset$, select $u(n) \in \mathcal{U}$ uniformly at random
Else Select group $\hat{h}(n)$ uniformly at random from $C(n)$ and run PIE(l) on $\mathcal{N}_{\hat{h}(n)}$
Compute: $\hat{\theta}_{ki}(n+1), \forall i, k$

derived for $K\times$PIE(l) when the topic is known, which means that the fact that the mapping h is unknown incurs a sub-logarithmic regret. Thus, our algorithm is asymptotically optimal since its regret upper bound matches the lower bound derived in Theorem 3.

6.1 Optimal Algorithms

To describe our algorithms, we introduce the following notations. Let $u(n)$ be the list selected in round n, and let $p_i(n)$ denote the position at which item i is shown if $i \in u(n)$, and $p_i(n) = 0$ otherwise. Let $X_{ki}(n) \in \{0, 1\}$ denote the relevance of item i when presented to a class-k user in round n. Define

$$o_{ki}(n) := \mathbf{1}\left\{k(n) = k, i \in u(n), \forall l' < p_i(n), X_{ku_{l'}(n)}(n) = 0\right\}$$

the event indicating whether a query of a class-k user arrives in round n and this user scans item i. Then we get a sample from θ_{ki} in round n iff $o_i(n) = 1$. Let $t_{ki}(n) := \sum_{n' \leq n} o_{ki}(n')$ be the number of samples obtained, up to round n, for θ_{ki}. The corresponding empirical mean is:

$$\hat{\theta}_{ki}(n) = \frac{1}{t_{ki}(n)} \sum_{n' \leq n} o_{ki}(n') X_{ki}(n')$$

if $t_{ki}(n) > 0$ and $\hat{\theta}_{ki}(n) = 0$ otherwise. The KL-UCB index $b_{ki}(n)$ of item i when presented to a class-k user in round n is:

$$b_{ki}(n) = \max\{q \in [0, 1] : t_{ki}(n) I(\hat{\theta}_{ki}(n), q) \leq f(n)\},$$

where $f(n) = \log(n) + 4\log(\log(n))$. Finally, for any user class k and topic h, we define $j_{kh}(n) = (j_{kh,1}(n), ..., j_{kh,N_h}(n))$, the items of $\mathcal{N}_h$ with empirical means sorted in decreasing order for users of class k in round n. Namely:

$$\hat{\theta}_{kj_{kh,1}(n)}(n) \geq \hat{\theta}_{kj_{kh,2}(n)}(n) \geq ... \geq \hat{\theta}_{kj_{kh,N_h}(n)}(n)$$

and $j_{kh,i}(n) \in \mathcal{N}_h$ for all k, h, and i.

The PIE-C(l, d) Algorithm. The algorithm is parametrized by $l \in [L]$ which indicates the position in which apparently suboptimal items are explored, and by d, a real number chosen strictly between δ and Δ. To implement such an algorithm, we do not need to know the maximum expected relevance δ of items of uninteresting topics, nor the lower bound Δ of the highest relevance of items whose topic corresponds to that of the query. We just need to know a number d in between.

In round n, PIE-C(l, d) maintains an estimator $\hat{h}(n)$ of the topic $h(k(n))$ requested by the user, and it proceeds as follows. Given the user-class $k(n)$, we first identify the set of *admissible* topics $C(n)$:

$$C(n) = \{h \in [K] : \max_{i \in \mathcal{N}_h} \hat{\theta}_{k(n)i}(n) \geq d\}.$$

This set corresponds to the topics that according to our observations up to round n, could be the topic requested by the class-$k(n)$ user.

(i) If $C(n) = \emptyset$, $\hat{h}(n) = -1$ (we don't know what the topic is), and we select $u(n)$ uniformly at random over the set of possible decisions $\mathcal{U}$;

(ii) If $C(n) \neq \emptyset$,

- Select $\hat{h}(n) \in C(n)$ uniformly at random;
- Define leaders at time n: $\mathcal{L}(n)$ lists in order the L items in $\mathcal{N}_{\hat{h}(n)}$ with largest empirical means,

$$\mathcal{L}(n) = (j_{k(n)\hat{h}(n),1}(n), ..., j_{k(n)\hat{h}(n),L}(n));$$

- Define the possible decisions $U_i(n)$ for all $i \in \mathcal{N}_{\hat{h}(n)} \setminus \mathcal{L}(n)$ obtained by replacing in $\mathcal{L}(n)$ the l-th item by i;
- Define

$$\mathcal{B}(n) = \{i \in \mathcal{N}_{\hat{h}(n)} \setminus \mathcal{L}(n) : b_{k(n)i}(n) \geq \hat{\theta}_{j_{k(n)\hat{h}(n),L}(n)}(n)\};$$

- (a) If $\mathcal{B}(n) = \emptyset$, select the list $\mathcal{L}(n)$, and (b) If $\mathcal{B}(n) \neq \emptyset$, choose $i(n)$ uniformly at random in $\mathcal{B}(n)$ and select either $\mathcal{L}(n)$ with probability $1/2$ or decision $U_{i(n)}(n)$ with probability $1/2$.

Note that when $\hat{h}(n)$ is believed to estimate $h(k(n))$ accurately (i.e., when $C(n) \neq \emptyset$), then the algorithm mimics the $K\times$PIE(l) algorithm. Refer to the pseudocode of PIE-C(l, d) for a formal description. The following theorem states that PIE-C(l, d) exhibits the same asymptotic regret as the optimal algorithms when the topic of each request is known.

THEOREM 5. *Assume that the reward function has the specific structure described in Case 1) (resp. 2)). For all $\delta < d < \Delta$, under the algorithm π =PIE-C(L, d) (resp. π =PIE-C$(1, d)$), we have for all θ:*

$$\limsup_{T \to \infty} \frac{R^\pi(T)}{\log(T)} \leq \sum_{k \in [K]} c_k(\theta).$$

6.2 Proof: Regret Upper Bound for PIE-C(l, d)

The proof of Theorem 5 consists in showing that the set of rounds at which the estimation of the topic $h(k(n))$ of the request fails is finite in expectation. As already mentioned, when the estimation is correct the algorithm behaves like $K\times$PIE(l), and the analysis of its regret in such rounds in the same as that under $K\times$PIE(l). Hence, we just need to control the size of the following set of rounds:

$$\mathcal{M} = \{n \geq 1 : \hat{h}(n) \neq h(k(n))\}.$$

LEMMA 2. *Under algorithm PIE-C(l,d) we have:*

$$\mathbb{E}[|\mathcal{M}|] \leq 2KN\left[2(N+1) + (d-\delta)^{-2} + (\Delta - d)^{-2}\right]$$

The above bound is minimized by setting $d = (\Delta + \delta)/2$, in which case:

$$\mathbb{E}[|\mathcal{M}|] \leq 4KN\left[N + 1 + 4(\Delta - \delta)^{-2}\right]$$

Proof. For all k, we define the most popular item for class-k users: $i_k^\star = \arg\max_i \theta_{ki}$. We decompose $\mathcal{M}$ by introducing the following sets:

$$\mathcal{M}_k = \{n \in \mathcal{M}, k(n) = k\},$$
$$\mathcal{M}_{k,-1} = \{n \in \mathcal{M}_k, \hat{h}(n) = -1, |\hat{\theta}_{ki_k^\star}(n) - \theta_{ki_k^\star}| \geq \Delta - d\},$$
$$\mathcal{M}_{k,i} = \{n \in \mathcal{M}_k, i = u_1(n), |\hat{\theta}_{ki}(n) - \theta_{ki}| \geq d - \delta\}.$$

$\mathcal{M}_{k,-1}$ is the set of rounds at which a user of class k makes a request, the set of admissible topics for class k users is empty $C(n)$, and $\theta_{ki_k^\star}$ is badly estimated. $\mathcal{M}_{k,i}$ is the set of rounds at which a user of class k makes a request, item $i \notin \mathcal{N}_{h(k)}$ is presented in the first slot (note that i is not interesting to that user) and θ_{ki} is badly estimated. We have that: $\mathcal{M} = \cup_{k=1}^{K} \mathcal{M}_k$ since $k(n) \in \{1, ..., K\}$.

We prove that for all k: $\mathcal{M}_k \subset \mathcal{M}_{k,-1} \cup (\cup_{i \notin \mathcal{N}_{h(k)}} \mathcal{M}_{k,i})$. Consider $n \in \mathcal{M}_k$, so that $k(n) = k$ and $\hat{h}(n) \neq h(k)$. We distinguish two cases:

(i) If $\hat{h}(n) = -1$, then $C(n) = \emptyset$. So $h(k) \notin C(n)$, and by definition of $C(n)$, this implies that $\max_{i \in \mathcal{N}_{h(k)}} \hat{\theta}_{ki}(n) \leq d$. Since $i_k^\star \in \mathcal{N}_{h(k)}$ we have $\hat{\theta}_{ki_k^\star}(n) \leq d$. Since $i_k^\star = \arg\max_i \theta_{ki}$, we have $\theta_{k,i_k^\star} \geq \Delta$. Hence we have that both $\hat{\theta}_{ki_k^\star}(n) \leq d$ and $\theta_{ki_k^\star} \geq \Delta$, so we have $|\hat{\theta}_{ki_k^\star}(n) - \theta_{ki_k^\star}| \geq \Delta - d$ and therefore $n \in \mathcal{M}_{k,-1}$.

(ii) If $\hat{h}(n) \notin \{h(k), -1\}$, then by design of the algorithm $u(n) \subset \{1, ..., N\} \setminus \mathcal{N}_{h(k)}$ since $\{\mathcal{N}_1, ..., \mathcal{N}_K\}$ forms a partition of $\{1, ..., N\}$. Hence there exists $i \notin \mathcal{N}_{h(k)}$ such that $u_1(n) = i$. By design of the algorithm, since $u_1(n) = i$, we have $\hat{\theta}_{ki}(n) = \arg\max_{i' \in \mathcal{N}_{\hat{h}(n)}} \hat{\theta}_{ki'}(n)$ and $\arg\max_{i' \in \mathcal{N}_{\hat{h}(n)}} \hat{\theta}_{ki'}(n) \geq d$ since $\hat{h}(n) \in C(n)$. Therefore $\hat{\theta}_{ki}(n) \geq d$ and we know that $\theta_{ki} \leq \delta$ since $i \notin \mathcal{N}_{h(k)}$, so that $|\hat{\theta}_{ki}(n) - \theta_{ki}| \geq d - \delta$. Summarizing, $\hat{h}(n) \notin \{h(k), -1\}$ implies that there exists $i \notin \mathcal{N}_{h(k)}$ such that $u_1(n) = i$ and $|\hat{\theta}_{ki}(n) - \theta_{ki}| \geq d - \delta$, therefore $n \in \cup_{i \notin \mathcal{N}_{h(k)}} \mathcal{M}_{k,i}$.

Hence we have proven, as announced, that $\mathcal{M}_k \subset \mathcal{M}_{k,-1} \cup_{i \notin \mathcal{N}_{h(k)}} \mathcal{M}_{k,i}$. We now upper bound the expected sizes of sets $\mathcal{M}_{k,-1}$ and $\mathcal{M}_{k,i}$.

Set $\mathcal{M}_{k,-1}$: When $n \in \mathcal{M}_{k,-1}$, $u(n)$ is uniformly distributed over the set of possible decisions $\mathcal{U}$, so that $\mathbb{P}[u_1(n) = i_k^\star | n \in \mathcal{M}_{k,-1}] = 1/N$. In turn, this implies that $\mathbb{E}[o_{ki_k^\star}(n) | n \in \mathcal{M}_{k,-1}] = 1/N$. Appying Lemma 5, second statement (with $H \equiv \mathcal{M}_{k,-1}$, $c \equiv 1$ and $\delta \equiv \Delta - d$), we obtain:

$$\mathbb{E}[|\mathcal{M}_{k,-1}|] \leq 2N \left[2N + (\Delta - d)^{-2}\right].$$

Set $\mathcal{M}_{k,i}$: When $n \in \mathcal{M}_{k,i}$, we have that $u_1(n) = i$ and $k(n) = k$ so that $\mathbb{E}[o_{k,i}(n) | n \in \mathcal{M}_{k,i}] = 1$. Applying Lemma 5, second statement (with $H \equiv \mathcal{M}_{k,i}$, $c \equiv 1$ and $\delta \equiv d - \delta$), we obtain:

$$\mathbb{E}[|\mathcal{M}_{k,i}|] \leq 2 \left[2 + (d - \delta)^{-2}\right].$$

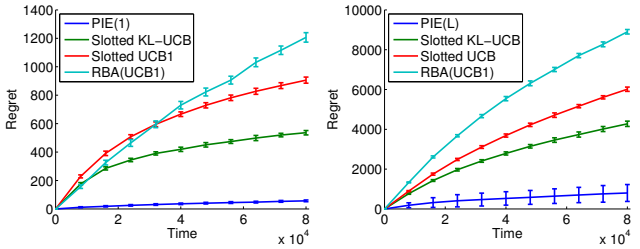
Using a union bound we have:

$$\begin{aligned}
\mathbb{E}[|\mathcal{M}_{k,-1}|] &\leq \mathbb{E}[|\mathcal{M}_{k,-1}|] + \sum_{i \notin \mathcal{N}_{h(k)}} \mathbb{E}[|\mathcal{M}_{k,i}|] \\
&\leq 2N \left[2N + (\Delta - d)^{-2}\right] + 2N \left[2 + (d - \delta)^{-2}\right] \\
&= 2N \left[2(N+1) + (d - \delta)^{-2} + (\Delta - d)^{-2}\right],
\end{aligned}$$

and summing over $k \in \{1, ..., K\}$ we obtain the announced result:

$$\begin{aligned}
\mathbb{E}[|\mathcal{M}|] &= \sum_{k=1}^{K} \mathbb{E}[|\mathcal{M}_k|] \\
&\leq 2KN \left[2(N+1) + (d - \delta)^{-2} + (\Delta - d)^{-2}\right],
\end{aligned}$$

which concludes the proof. □



(a) Case 2: $\forall l, r(l) = 1$. (b) Case 1: $\forall l, r(l) = 2^{1-l}$.

Figure 1: Performance of PIE(1) / PIE(L) and other UCB-based algorithms. A single group of items and users. Error bars represent the standard deviation.

7. NUMERICAL EXPERIMENTS

In this section, we evaluate the practical performance of our algorithms using both artificially generated and real-world data[1].

7.1 Artificial Data

We first evaluate the PIE and PIE-C algorithms in the scenarios presented in Sections 4, 5, and 6. In these scenarios, the algorithms are optimal and hence they should outperform any other algorithm.

A Single group of users / items. First we assume there exists only one relevant topic ($K = 1$) consisting of $N = 800$ items. We consider $L = 10$ and evaluate the performance of the algorithms over the arrival of $T = 8 \times 10^4$ user queries. The parameter θ is artificially generated as follows:

$$\theta_i = 0.55 \times (1 - (i-1)/(N-1)).$$

In Figure 1(a), we consider the reward to be $r(l) = 1$ for $l \in \{1, ..., L\}$ while in Figure 1(b), we assume the reward decreases geometrically with the slot ($r(l) = 2^{1-l}$, $l \in \{1, ..., L\}$). Under these assumptions, PIE(1) and PIE(L) respectively are asymptotically optimal according to Theorem 2. We compare their performance to that of Slotted UCB, Slotted KL-UCB algorithms, and RBA (Ranked Bandit Algorithm) proposed in [7] and [11]. In Slotted UCB (resp. KL-UCB), the L items with the largest UCB (resp. KL-UCB) indexes are displayed, whereas RBA runs L independent bandit algorithms, one for each slot. In particular, for all items k, the bandit algorithm assigned to slot l can only access the observations obtained from k when k was played in slot l (RBA attempts to learn an item's so-called *marginal utility* for each slot). Observe that PIE significantly outperforms all other algorithms.

Multiple groups of users / items. Next, we consider $K = 5$ groups of users and items, and $N = 4,000$ items. We assume all groups are of equal size so that $\phi_k = 1/K$ for all k. There are N/K items in each group. We define $j(i,k) = (i - h(k)N/K)$, and generate the parameter θ as follows:

$$\theta_{ki} = \begin{cases} 0.55 \times (1 - (j(i,k) - 1)/(N-1)) & \text{if } i \in \mathcal{N}_{h(k)}, \\ 0.05 & \text{otherwise.} \end{cases}$$

Figure 2 presents the performance of $5 \times$PIE(1) (referred to as PIE(1) in the figure) when the decision maker knows the mapping between user classes and topics $h(\cdot)$, and that of PIE-C(1,0.5) when $h(\cdot)$ is unknown. Figure 2 corroborates the theoretical result of Theorem 5: The performance loss due to the need to learn the mapping $h(\cdot)$ is rather limited, especially the time horizon grows large.

[1]We use the Movielens10M dataset, available at http://grouplens.org/datasets/movielens/

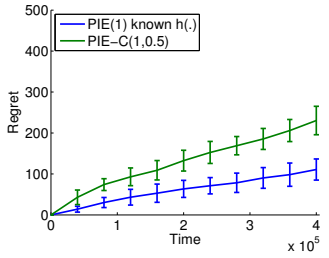
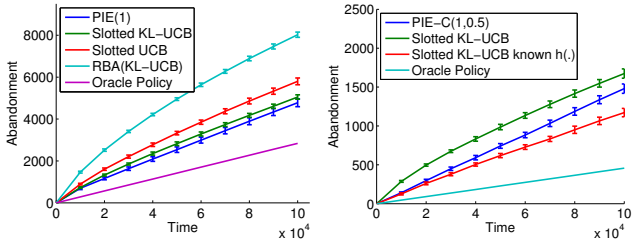


Figure 2: Performance of PIE(1) and PIE-C(1,d). $K = 5$ groups of users and items. Case 2: $\forall l$, $r(l) = 1$.



(a) $K = 1$ topic. (b) $K = 4$ topics.

Figure 3: Performance of PIE(1) and PIE-C$(1, d)$ on real world data.

7.2 Real-world Data

We further investigate the performance of our algorithms on real-world systems. We use the Movielens dataset which contains the ratings given by users to a large set of movies. The dataset is a large matrix $X = (X_{a,m})$ where $X_{a,m} \in \{0, 1, ..., 5\}$ is the rating given by user a to movie m. The highest rating is 5, the lowest is 1, and 0 denotes an absence of rating, as most users did not watch the whole set of movies. From matrix X, we created a binary matrix Y such that $Y_{a,m} = 0$ if $X_{a,m} < 4$ and $Y_{a,m} = 1$ otherwise. We say that movie m is *interesting* to user a iff $Y_{a,m} = 1$.

We first selected the 100 most popular movies with less than 13,000 ratings (to avoid movies with good ratings for a large majority of users) and the 61,357 users who rated at least one of those movies. We extracted the corresponding sub-matrix of Y. To cluster the users and the movies, we use the classical spectral method. We extracted the 4 largest singular values of Y and their corresponding singular vectors $\gamma_i, i \in \{1, 2, 3, 4\}$. We then assigned each user a to the cluster $k = \text{argmax}_i\, Y_a \cdot \gamma_i$, where Y_a is the a-th line of matrix Y. We performed a similar classification of movies.

In Figure 3(a), we consider class-1 users, and compare the performance of algorithms already considered in Subsection 7.1, Scenario 1. The simulation proceeds as follows: in round n, we draw a class-1 user, denoted by $a(n)$, uniformly at random. The considered algorithm chooses an action $u(n)$, if $u(n)$ contains an interesting movie i for user $a(n)$, i.e., $Y_{a(n)i} = 1$, the system collects a unit reward, otherwise the reward is 0. We emulate the semi-bandit feedback by assuming the algorithm is informed about uninteresting movies j, i.e., $Y_{a(n)j} = 0$, placed above i in the list of $L = 10$ movies. Here the performance of the algorithm is quantified through the notion of *abandonment*, as introduced in [7]. The abandonment is the number of rounds in which no interesting movies are displayed. As a benchmark, we use an Oracle policy that displays the L most popular movies for users of class 1 in every round. Note that we use abandonment as a performance metric rather than the regret, because the optimal policy is hard to compute given the fact that the ratings offered by a user to different movies are not always independent in our data set. Again, PIE outperforms the Slotted variants of UCB and KL-UCB which in turn significantly outperform RBA(KL-UCB). In fact, the cost of learning in PIE (compared to the Oracle policy is limited): the abandonment under PIE does not exceed twice that of the Oracle policy. Note that the performance gain under PIE compared to Slotted KL-UCB is much higher in our artificial data simulations. We believe that this may be firstly due to the inaccuracy of our model when used against this particular data-set, and secondly due to the fact that the gain under PIE increases with the number of items N.

In Figure 3(b), we consider $K = 4$ groups, each topic consisting of 25 items. Again, the performance of PIE-C algorithm is not too far from that of the Oracle policy. PIE-C is compared to Slotted KL-UCB and a Slotted KL-UCB aware of the groups and of the mapping $h(\cdot)$. The former just ignores the group structure and runs as if there were a single group only, whereas the latter consists in K parallel and independent instances of Slotted KL-UCB, one for each user class k and item group $h(k)$. PIE-C outperforms Slotted KL-UCB, and its performance is similar to that of Slotted KL-UCB with known mapping $h(\cdot)$. Again this indicates that PIE-C rapidly learns the mapping $h(\cdot)$.

8. CONCLUSION

In this paper, we investigated the design of learning-to-rank algorithms for online systems, such as search engines and ad-display systems. We proposed PIE and PIE-C, two asymptotically optimal algorithms that rapidly learn users' preferences, and the most relevant items to be listed in response to user queries. These two algorithms are devised assuming that users and items are clustered, and that the decision maker knows the class of the user issuing the query. It would be interesting to further extend these algorithms to scenarios where the classes of the various users are initially unknown. The paper also presents a preliminary performance evaluation of our algorithms. In future work, we will further investigate the way our algorithms perform against various kinds of real-world dataset, including hopefully real traces extracted from search engines, such as google or bing.

9. ACKNOWLEDGEMENTS

Research supported by SSF, VR and ERC grant 308267.

10. REFERENCES

[1] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender, "Learning to rank using gradient descent," in *Proc. of ICML*, 2005.

[2] S. Pandey, D. Agarwal, D. Chakrabarti, and V. Josifovski, "Bandits for taxonomies: A model based approach," in *Proc. of SIAM SDM*, 2007.

[3] F. Radlinski and T. Joachims, "Active exploration for learning rankings from clickthrough data," in *Proc. of ACM SIGKDD*, 2007.

[4] M. J. Streeter, D. Golovin, and A. Krause, "Online learning of assignments." in *Proc. of NIPS*, 2009.

[5] H. Robbins, "Some aspects of the sequential design of experiments," *Bulletin of the American Mathematical Society*, vol. 58, no. 5, pp. 527–535, 1952.

[6] J. Gittins, *Bandit Processes and Dynamic Allocation Indices*. John Wiley, 1989.

[7] F. Radlinski, R. Kleinberg, and T. Joachims, "Learning diverse rankings with multi-armed bandits," in *Proc. of ICML*, 2008.

[8] Y. Yue and T. Joachims, "Interactively optimizing information retrieval systems as a dueling bandits problem," in *Proc. of ICML*, 2009.

[9] Y. Yue and C. Guestrin, "Linear submodular bandits and their application to diversified retrieval," in *Proc. of NIPS*, 2011.

[10] S. Khuller, A. Moss, and J. S. Naor, "The budgeted maximum coverage problem," *Inf. Process. Lett.*, vol. 70, no. 1, pp. 39–45, Apr. 1999.

[11] P. Kohli, M. Salek, and G. Stoddard, "A fast bandit algorithm for recommendations to users with heterogeneous tastes," in *Proc. of AAAI*, 2013.

[12] S. Agrawal, Y. Ding, A. Saberi, and Y. Ye, "Correlation robust stochastic optimization," in *Proc. of ACM SODA*, 2010.

[13] A. Slivkins, F. Radlinski, and S. Gollapudi, "Ranked bandits in metric spaces: learning optimally diverse rankings over large document collections," *Journal of Machine Learning Research*, 2013.

[14] S. Bubeck and N. Cesa-Bianchi, "Regret analysis of stochastic and nonstochastic multi-armed bandit problems," *Foundations and Trends in Machine Learning*, vol. 5, no. 1, pp. 1–122, 2012.

[15] R. Agrawal, "The continuum-armed bandit problem," *SIAM J. Control and Optimization*, vol. 33, no. 6, pp. 1926–1951, 1995.

[16] R. Kleinberg, A. Slivkins, and E. Upfal, "Multi-armed bandits in metric spaces," in *Proc. of STOC*, 2008.

[17] S. Bubeck, R. Munos, G. Stoltz, and C. Szepesvári, "Online optimization in x-armed bandits," in *Proc. of NIPS*, 2008.

[18] S. Magureanu, R. Combes, and A. Proutiere, "Lipschitz bandits: Regret lower bound and optimal algorithms," in *Proc. of COLT*, 2014.

[19] V. Dani, T. P. Hayes, and S. M. Kakade, "Stochastic linear optimization under bandit feedback," in *Proc. of COLT*, 2008.

[20] A. Flaxman, A. T. Kalai, and H. B. McMahan, "Online convex optimization in the bandit setting: gradient descent without a gradient," in *Proc. of ACM SODA*, 2005.

[21] L. Bui, R. Johari, and S. Mannor, "Clustered bandits," *http://arxiv.org/abs/1206.4169*, 2012.

[22] T. L. Lai and H. Robbins, "Asymptotically efficient adaptive allocation rules," *Advances in Applied Mathematics*, vol. 6, no. 1, pp. 4–2, 1985.

[23] T. L. Graves and T. L. Lai, "Asymptotically efficient adaptive choice of control laws in controlled markov chains," *SIAM Journal on Control and Optimization*, vol. 35, no. 3, pp. 715–743, 1997.

[24] A. Garivier and O. Cappé, "The KL-UCB algorithm for bounded stochastic bandits and beyond," in *Proc. of COLT*, 2011.

[25] R. Combes and A. Proutiere, "Unimodal bandits: Regret lower bounds and optimal algorithms," in *Proc. of ICML, http://arxiv.org/abs/1405.5096*, 2014.

APPENDIX

A. SUPPORTING LEMMAS FOR THE PROOF OF THEOREM 2

Lemma 3 allows to control the fluctuations of the estimate $\hat{\theta}_i(n)$ evaluated at a random time ϕ. We assume that ϕ is a stopping time, and that the number of rounds before ϕ where a decision containing i has been taken is greater than a number s. This result is instrumental in analyzing the finite time regret of algorithms (such as ours) that take decisions based on the estimates $\hat{\theta}_i(n)$. Lemma 3 is a consequence of Lemma 4, which is reproduced here for completeness.

LEMMA 3. *Let $\{Z_t\}_{t\in\mathbb{Z}}$ be a sequence of independent random variables with values in $[0,1]$. Define $\mathcal{F}_n$ the σ-algebra generated by $\{Z_t\}_{t\leq n}$ and the filtration $\mathcal{F}=(\mathcal{F}_n)_{n\in\mathbb{Z}}$. Consider $s\in\mathbb{N}$, $n_0\in\mathbb{Z}$ and $T\geq n_0$. We define $S_n=\sum_{t=n_0}^{n} B_t(Z_t-\mathbb{E}[Z_t])$, where $B_t\in\{0,1\}$ is a $\mathcal{F}_{t-1}$-measurable random variable. Further consider that for all t, almost surely we have $B_t\geq \bar{B}_t C_t$, where both $\bar{B}_t$ and C_t are $\{0,1\}$ -valued, $\mathcal{F}_{t-1}$-measurable random variables, such that for all t: $\mathbb{P}[C_t=1]\geq c>0$.*

Further define $t_n=\sum_{t=n_0}^{n} B_t$ and $c_n=\sum_{t=n_0}^{n}\bar{B}_t$. Define $\phi\in\{n_0,\dots,T+1\}$ a $\mathcal{F}$-stopping time such that either $c_\phi\geq s$ or $\phi=T+1$.

Then for all $\epsilon>0$ we have that:

$$\mathbb{P}[S_\phi\geq t_\phi\delta\ ,\ \phi\leq T]\leq e^{-2s\epsilon^2c^2}+e^{-2c(1-\epsilon)s\delta^2}.$$

As a consequence:

$$\mathbb{P}[|S_\phi|\geq t_\phi\delta\ ,\ \phi\leq T]\leq 2(e^{-2s\epsilon^2c^2}+e^{-2c(1-\epsilon)s\delta^2}).$$

Proof. We prove the first statement, as the second statement follows by symmetry.

When event $S_\phi\geq t_\phi\delta$ occurs, we have either that (a) $t_\phi\leq c(1-\epsilon)c_\phi$ or (b) $S_\phi\geq t_\phi\delta$ and $t_\phi\geq c(1-\epsilon)c_\phi\geq c(1-\epsilon)s$. In case (a), if $\phi\leq T$, we have:

$$\sum_{t=n_0}^{n}\bar{B}_tC_t\leq\sum_{t=n_0}^{n}B_t=t_\phi\leq c(1-\epsilon)c_\phi=c(1-\epsilon)\sum_{t=n_0}^{\phi}\bar{B}_t,$$

and therefore:

$$\sum_{t=n_0}^{\phi}\bar{B}_tC_t\leq c(1-\epsilon)\sum_{t=n_0}^{\phi}\bar{B}_t$$
$$\sum_{t=n_0}^{\phi}\bar{B}_t(C_t-c)\leq -c\epsilon\sum_{t=n_0}^{\phi}\bar{B}_t$$
$$\sum_{t=n_0}^{\phi}\bar{B}_t(C_t-\mathbb{E}[C_t])\leq -c\epsilon\sum_{t=n_0}^{\phi}\bar{B}_t.$$

where the last inequality holds because $\mathbb{E}[C_t]\geq c$ for all t. We may now apply Lemma 4 (with $Z_t\equiv C_t$, $B_t\equiv\bar{B}_t$, and $\delta\equiv c\epsilon$) to obtain:

$$\begin{aligned}&\mathbb{P}[t_\phi\leq c(1-\epsilon)c_\phi\ ,\ \phi\leq T]\\&\quad\leq\mathbb{P}\left[\sum_{t=n_0}^{\phi}\bar{B}_t(C_t-\mathbb{E}[C_t])\leq -c\epsilon\sum_{t=n_0}^{\phi}\bar{B}_t\ ,\ \phi\leq T\right]\\&\quad\leq e^{-2s\epsilon^2c^2}.\end{aligned}$$

In case (b), define another stopping time ϕ', such that $\phi'=\phi$ if $t_\phi\geq c(1-\epsilon)c_\phi$ and $\phi'=T+1$ otherwise. Note that ϕ' is indeed a stopping time. We apply Lemma 4 a second time (with $\phi\equiv\phi'$, $s\equiv c(1-\epsilon)s$) to obtain:

$$\begin{aligned}&\mathbb{P}[S_\phi\geq t_\phi\delta\ ,\ t_\phi\geq c(1-\epsilon)c_\phi\ ,\ \phi\leq T]\\&\quad=\mathbb{P}\left[S_{\phi'}\geq t_{\phi'}\delta\ ,\ \phi'\leq T\right]\leq e^{-2c(1-\epsilon)s\delta^2}.\end{aligned}$$

Summing the inequalities obtained in cases (a) and (b), we prove the announced result:

$$\mathbb{P}[S_\phi\geq t_\phi\delta\ ,\ \phi\leq T]\leq e^{-2s\epsilon^2c^2}+e^{-2c(1-\epsilon)s\delta^2}.$$

which concludes the proof. □

LEMMA 4. *([25]) Let $\{Z_t\}_{t\in\mathbb{Z}}$ be a sequence of independent random variables with values in $[0,1]$. Define $\mathcal{F}_n$ the σ-algebra generated by $\{Z_t\}_{t\leq n}$ and the filtration $\mathcal{F}=(\mathcal{F}_n)_{n\in\mathbb{Z}}$. Consider $s\in\mathbb{N}$, $n_0\in\mathbb{Z}$ and $T\geq n_0$. We define $S_n=\sum_{t=n_0}^{n}B_t(Z_t-\mathbb{E}[Z_t])$, where $B_t\in\{0,1\}$ is a $\mathcal{F}_{t-1}$-measurable random variable. Further define $t_n=\sum_{t=n_0}^{n}B_t$. Define $\phi\in\{n_0,\dots,T+1\}$ a $\mathcal{F}$-stopping time such that either $t_\phi\geq s$ or $\phi=T+1$.*

Then we have that:

$$\mathbb{P}[S_\phi\geq t_\phi\delta\ ,\ \phi\leq T]\leq\exp(-2s\delta^2).$$

As a consequence:

$$\mathbb{P}[|S_\phi|\geq t_\phi\delta\ ,\ \phi\leq T]\leq 2\exp(-2s\delta^2).$$

Lemma 5 is a consequence of Lemma 3, and allows to upper bound the size of random sets of rounds where decisions containing i have been sampled and the empirical mean $\hat{\theta}_i(n)$ deviates from its expectation by more than a fixed amount $\delta>0$.

LEMMA 5. *Let us fix $c>0$ and $1\leq i\leq N$. Consider a random set of rounds $H\subset\mathbb{N}$, such that, for all n, $\mathbf{1}\{n\in H\}$ is $\mathcal{F}_{n-1}$ measurable. Further assume for all n we have: $\mathbb{E}[o_i(n)|n\in H]\geq c>0$. Consider a random set $\Lambda=\cup_{s\geq 1}\{\tau_s\}\subset\mathbb{N}$, where for all s, τ_s is a stopping time such that $\sum_{n=1}^{\tau_s}\mathbf{1}\{n\in H\}\geq s$.*

Then for all i and $\epsilon>0$ and $\delta>0$ we have that:

$$\sum_{n\geq 0}\mathbb{P}[n\in\Lambda,|\hat{\theta}_i(n)-\theta_i|\geq\delta]\leq c^{-1}\left[\frac{1}{\epsilon^2c}+\frac{1}{\delta^2(1-\epsilon)}\right].$$

As a consequence:

$$\sum_{n\geq 0}\mathbb{P}[n\in\Lambda,|\hat{\theta}_i(n)-\theta_i|\geq\delta]\leq 2c^{-1}\left[2c^{-1}+\delta^{-2}\right].$$

Proof. Fix $T<\infty$ and s. Apply Lemma 3 (with $Z_t\equiv X_i(t)$, $B_t\equiv o_i(n)$, $\bar{B}_t\equiv\mathbf{1}\{n\in H\}$, C_t a Bernoulli variable with parameter $\mathbb{E}[o_i(n)|n\in H]$ which is conditionally independent of $\mathbf{1}\{n\in H\}$) to obtain:

$$\mathbb{P}[|\hat{\theta}_i(\tau_s)-\theta_i|\geq\delta,\tau_s\leq T]\leq 2(e^{-2s\epsilon^2c^2}+e^{-2c(1-\epsilon)s\delta^2}).$$

Using a union bound over s, for all $\epsilon>0$ we get:

$$\begin{aligned}\sum_{n\leq T}\mathbb{P}[n\in\Lambda,|\hat{\theta}_i(n)-\theta_i|\geq\delta]&\leq\sum_{s\geq 1}\mathbb{P}[|\hat{\theta}_i(\tau_s)-\theta_i|\geq\delta,\tau_s\leq T]\\&\leq\sum_{s\geq 1}2(e^{-2s\epsilon^2c^2}+e^{-2c(1-\epsilon)s\delta^2})\\&\leq\frac{1}{\epsilon^2c^2}+\frac{1}{c(1-\epsilon)\delta^2}\\&=c^{-1}\left[\frac{1}{\epsilon^2c}+\frac{1}{\delta^2(1-\epsilon)}\right],\end{aligned}$$

where we have used the following inequality twice $\sum_{s\geq 1}e^{-sw}\leq\int_0^{+\infty}e^{-sw}ds=1/w$, valid for all $w>0$. Since

the above inequality holds for all T, and its r.h.s. does not depend on T we conclude that:

$$\sum_{n\geq 1}\mathbb{P}[n\in\Lambda,|\hat{\theta}_i(n)-\theta_i|\geq\delta]\leq c^{-1}\left[\frac{1}{\epsilon^2 c}+\frac{1}{\delta^2(1-\epsilon)}\right],$$

which concludes the proof of the first statement.

The second statement is obtained by setting $\epsilon=1/2$. □

COROLLARY 1. *Consider $c>0$ and $1\leq i\leq N$ fixed. Consider a random set of instants $H\subset\mathbb{N}$, such that, for all n, $\mathbf{1}\{n\in H\}$ is $\mathcal{F}_{n-1}$ measurable. Further assume for all n we have: $\mathbb{E}[o_i(n)|n\in H]\geq c>0$. Define $h_i(n)=\sum_{n'\leq n}\mathbf{1}\{n'\in H\}$. Consider $\epsilon>0$ and $\delta>0$ and define the set:*

$$\mathcal{H}=\left\{n\in H:\left(t_i(n)\leq(1-\epsilon)h_i(n)\right)\vee\left(|\hat{\theta}_i(n)-\theta_i|\geq\delta\right)\right\}$$

Then we have:

$$\mathbb{E}[|\mathcal{H}|]\leq c^{-1}\left[c^{-1}\epsilon^{-2}+\delta^{-2}(1-\epsilon)^{-1}\right].$$

Proof. Straightforward from the proof of Lemma 5 with $\Lambda=H$. □

Lemma 6 is a straightforward consequence of Theorem 10 in [24], and states that the expected number of times the index of a given item i underestimates its true value is finite, and upper bounded by a constant that does not depend on the parameters $(\theta_i)_i$.

LEMMA 6. *([24]) Define:*

$$b_i(n)=\max\{q\in[0,1]:t_i(n)I(\hat{\theta}_i(n),q)\leq f(n)\},$$

with $f(n)=\log(n)+4\log(\log(n))$.

There exists a constant C_0 independent of $(\theta_i)_i$ such that for all i we have:

$$\sum_{n\geq 0}\mathbb{P}[b_i(n)<\theta_i]\leq C_0.$$

In particular one has $C_0\leq 2e\sum_{n\geq 1}\lceil f(n)\log(n)\rceil e^{-f(n)}\leq 15$.