Best Arm Identification for Cascading Bandits in the Fixed Confidence Setting

Zixin Zhong [1] **Wang Chi Cheung** [2 3] **Vincent Y. F. Tan** [1 3 4]

Abstract

We design and analyze CASCADEBAI, an algorithm for finding the best set of K items, also called an arm, within the framework of cascading bandits. An upper bound on the time complexity of CASCADEBAI is derived by overcoming a crucial analytical challenge, namely, that of probabilistically estimating the amount of available feedback at each step. To do so, we define a new class of random variables (r.v.'s) which we term as left-sided sub-Gaussian r.v.'s; this class is a relaxed version of the sub-Gaussian r.v.'s. This enables the application of a sufficiently tight Bernstein-type concentration inequality. We show, through the derivation of a lower bound on the time complexity, that the performance of CASCADEBAI is optimal in some practical regimes. Finally, extensive numerical simulations corroborate the efficacy of CASCADEBAI as well as the tightness of our upper bound on its time complexity.

1. Introduction

Online recommender systems seek to recommend a small list of items (such as movies or hotels) to users based on a larger ground set $[L] := \{1, \ldots, L\}$ of items. In this paper, we consider the *cascading bandits* model (Craswell et al., 2008; Kveton et al., 2015a), which is widely used in information retrieval and online advertising. Upon seeing the chosen list, the user looks at the items sequentially. She *clicks* on an item if she is *attracted* by it and skips to the next one otherwise. This process stops when she clicks on one item in the list or if no item is clicked, it is deemed that she is *not attracted* by *any* of the items. The items that are in the ground set but not in the chosen list and those in the list that come after the attractive one are *unobserved*.

Each item $i \in [L]$, with a certain *click probability* $w(i) \in [0, 1]$ which is *unknown* to the learning agent, attracts the user independently of other items. Under this assumption, the optimal solution is the list of items with largest $w(i)$'s. Based on the chosen lists and obtained feedback in previous steps, the agent tries to learn the click probabilities (explore the combinatorial space) in order to find the optimal list with high probability in as few time steps as possible.

Main Contributions. Given $\delta > 0$, a learning agent aims to find a list of optimal items of size K with probability at least $1 - \delta$ in minimal time steps. To achieve a greater generality, we provide results for identifying a list of near-optimal items (Even-Dar et al., 2002; Mannor & Tsitsiklis, 2004; Kalyanakrishnan et al., 2012), where the notion of near-optimality is precisely defined in Section 2. First, we design CASCADEBAI(ϵ, δ, K) and derive an upper bound on its time complexity. Second, we establish a lower bound on the time complexity of *any* best arm identification (BAI) algorithm in cascading bandits, which implies that the performance of CASCADEBAI(ϵ, δ, K) is optimal in some regimes. Finally, our extensive numerical results corroborate the efficacy of CASCADEBAI(ϵ, δ, K) and the tightness of our upper bound on its time complexity.

Different from combinatorial semi-bandit settings, the amount of feedback in cascading bandits is, in general, random. The analysis of cascading bandits involves the unique challenge in adapting to the variation of the amount of feedback across time. To this end, we define a random variable (r.v.) that describes the feedback from the user at a step and bound its expectation. We define a novel class of r.v.'s, known as *left-sided sub-Gaussian* (LSG) r.v.'s, and apply a concentration inequality to quantify the variation of the amount of feedback.

Bernstein-type concentration inequalities are applied in many stochastic bandit problems and indicate that sub-Gaussian (SG) distributions possess light tails (Audibert & Bubeck, 2010). Since it turns out that we only need to analyze a one-sided tail in this work, it suffices to consider a one-sided SG condition, which motivates the definition

[1]Department of Mathematics, National University of Singapore, Singapore [2]Department of Industrial Systems and Management, National University of Singapore, Singapore [3]Institute of Operations Research and Analytics, National University of Singapore, Singapore [4]Department of Electrical and Computer Engineering, National University of Singapore, Singapore. Correspondence to: Zixin Zhong <zixin.zhong@u.nus.edu>, Wang Chi Cheung <isecwc@nus.edu.sg>, Vincent Y. F. Tan <vtan@nus.edu.sg>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020.

of LSG. We also provide a general estimate of a certain corresponding parameter in Theorem 5.4, which is crucial for the utilization of the inequality. This may be of independent interest. Summary and future work are deferred to Appendix 7.

Literature review. In a stochastic combinatorial bandit (SCB) model, an arm corresponds to a list of items in the ground set, and each item is associated with an r.v. at each time step. The corresponding reward depends on the constituent items' realizations. We first review the related works on the BAI problem, in which a learning agent aims to identify an *optimal arm*, i.e., a list of optimal items. (i) Given $\delta > 0$, a learning agent aims to identify an optimal arm with probability $1-\delta$ in minimal time steps (Jamieson & Nowak, 2014; Kalyanakrishnan et al., 2012). (ii) Given $B > 0$, an agent aims to maximize the probability of identifying an optimal arm in B steps (Auer et al., 2002; Audibert & Bubeck, 2010; Carpentier & Locatelli, 2016). These two settings are known as the *fixed-confidence* and *fixed-budget* setting respectively. Under the fixed-confidence setting, the early works aim to identify only one optimal item (Audibert & Bubeck, 2010) and the later ones aim to find an optimal arm (Chen et al., 2014; Rejwan & Mansour, 2019). Besides, Mannor & Tsitsiklis (2004); Kaufmann et al. (2016); Agarwal et al. (2017) provide problem-dependent lower bounds on the time complexity when Kalyanakrishnan et al. (2012) establishes a problem-independent one. All these existing works above pertain to the *semi-bandit feedback* setting, where an agent observes realizations of *all* pulled items. Finally, we would like to highlight Kuroki et al. (2019) and Rejwan & Mansour (2020) who consider the *full-bandit feedback* setting, where an agent only observes the sums of the realizations of all pulled items.

Secondly, we review the relevant works on the *regret minimization* (RM) problem, in which an agent aims to maximize his overall reward, or equivalently to minimize the so-called *cumulative regret*. Under the semi-bandit feedback setting, this problem has been extensively studied by Lai & Robbins (1985); Anantharam et al. (1987); Kveton et al. (2014); Li et al. (2010); Qin et al. (2014). Moreover, motivated by numerous applications in clinical analysis and online advertisement, some researchers consider SCB models with *partial feedback*, where an agent observes realizations of only a portion of pulled items. One prime model that incorporates the partial feedback is cascading bandits (Craswell et al., 2008; Kveton et al., 2015a). Recently, Kveton et al. (2015b); Li et al. (2016); Zong et al. (2016); Wang & Chen (2017); Cheung et al. (2019) studied this model and derived various regret bounds.

When the RM problem is studied with both semi-bandit and partial feedback, the BAI problem has only been studied in the semi-bandit feedback setting thus far. Despite existing works, analysis of the BAI problem in the more challenging case of partial feedback is yet to be done. Our work fills in this gap in the literature by studying the fixed-confidence setting in cascading bandits, and our analysis provides tools for handling the statistical dependence between the amount of feedback and that of time steps in the cascading bandit setting.

2. Problem Setup

For brevity, we denote the set $\{1, \dots, n\}$ by $[n]$ for any $n \in \mathbb{N}$, and the set of all m-permutations of $[n]$, i.e., all ordered m-subsets of $[n]$, by $[n]^{(m)}$ for any $m \leq n$. Let there be $L \in \mathbb{N}$ ground items, contained in $[L]$. Each item $i \in [L]$ is associated with a *weight* $w(i) \in [0,1]$, signifying the item's click probability. We define an *arm* as a list of $K \leq L$ items in $[L]^{(K)}$. At each time step t, the agent pulls an arm $S_t := (i_1^t, \dots, i_K^t) \in [L]^{(K)}$. Then the user examines the items from i_1^t to i_K^t one at a time, until one item is clicked or all items are examined. For each item $i \in [L]$, $W_t(i) \sim \text{Bern}(w(i))$ are i.i.d. across time. The agent observes $W_t(i) = 1$ iff the user clicks on i. The *feedback* $\boldsymbol{O}_t$ from the user is defined as a vector in $\{0, 1, \star\}^K$, where $0, 1, \star$ represents observing no click, observing a click and no observation respectively. For example, if $K = 4$ and the user clicks on the third item at time step 2, we have $\boldsymbol{O}_2 = \{0, 0, 1, \star\}$. Clearly, there is a one-to-one mapping from $\boldsymbol{O}_t$ to the integer

$$\tilde{k}_t := \min\{1 \leq k \leq K : W_t(i_k^t) = 1\},$$

where we assume $\min \emptyset = \infty$. If $\tilde{k}_t < \infty$ (i.e., $\boldsymbol{O}_t$ is not the all-zero vector), the agent observes $W_t(i_k^t) = 0$ for $1 \leq k < \tilde{k}_t$, and also observes $W_t(i_{\tilde{k}_t}^t) = 1$, but does not observe $W_t(i_k^t)$ for $k > \tilde{k}_t$. Otherwise, we have $\tilde{k}_t = \infty$ (i.e., $\boldsymbol{O}_t$ is the all-zero vector), then the agent observes $W_t(i_k^t) = 0$ for $1 \leq k \leq K$. We denote $\bar{w}(i) = 1 - w(i)$, $\boldsymbol{w} = (w(1), \dots, w(L))$, and the probability law (resp. the expectation) of the process $(\{W_t(i)\}_{i,t})$ by $\mathbb{P}_{\boldsymbol{w}}$ (resp. $\mathbb{E}_{\boldsymbol{w}}$).

Without loss of generality, we assume $w^* := w(1) \geq w(2) \geq \dots \geq w(L) := w'$. We say item i is *optimal* if $w(i) \geq w(K)$. We assume $w(K) > w(K+1)$ to ensure there are exactly K optimal items. Next, we say item i is ϵ-*optimal* ($\epsilon \geq 0$) if $w(i) \geq w(K) - \epsilon$ and set $K'_\epsilon := \max\{i \in [L] : w(i) \geq w(K) - \epsilon\}$. Then $[K'_\epsilon]$ is the set of all ϵ-optimal items, $[K]^{(K)}$ is the set of all *optimal arms* S^* (up to permutation), and $[K'_\epsilon]^{(K)}$ is the set of all ϵ-*optimal arms*.

To identify an ϵ-optimal arm, an agent uses an *algorithm* π that decides which arms to pull, when to stop pulling, and which arm $\hat{S}^\pi$ to choose eventually. A deterministic and non-anticipatory online algorithm consists in a triple $\pi := ((\pi_t)_t, \mathcal{T}^\pi, \phi^\pi)$ in which:

• the *sampling rule* π_t determines, based on the observation

history, the arm S_t^π to pull at time step t; in other words, S_t^π is $\mathcal{F}_{t-1}$-measurable, with $\mathcal{F}_t := \sigma(S_1^\pi, \boldsymbol{O}_1^\pi, \ldots, S_t^\pi, \boldsymbol{O}_t^\pi)$;
• the *stopping rule* determines the termination of the algorithm, which leads to a *stopping time* $\mathcal{T}^\pi$ with respect to $(\mathcal{F}_t)_{t\in\mathbb{N}}$ satisfying $\mathbb{P}(\mathcal{T}^\pi < +\infty) = 1$;
• the *recommendation rule* ϕ^π chooses an arm $\hat{S}^\pi$, which is $\mathcal{F}_{\mathcal{T}^\pi}$-measurable.

We define the *time complexity* of π as $\mathcal{T}^\pi$. Under the fixed-confidence setting, a risk parameter (failure probability) $\delta \in (0,1)$ is fixed. We say an algorithm π is (ϵ,δ,K)-*PAC (probably approximately correct)* if $\mathbb{P}_{\boldsymbol{w}}(\hat{S}^\pi \subset [K'_\epsilon]) \geq 1-\delta$. The goal is to obtain an (ϵ,δ,K)-PAC algorithm π such that $\mathbb{E}_{\boldsymbol{w}}\mathcal{T}^\pi$ is small and $\mathcal{T}^\pi$ is small with high probability. We also define the *optimal expected time complexity* over all (ϵ,δ,K)-PAC algorithms as

$$\mathbb{T}^*(\boldsymbol{w},\epsilon,\delta,K) := \inf\{\mathbb{E}_{\boldsymbol{w}}\mathcal{T}^\pi : \pi \text{ is } (\epsilon,\delta,K)\text{-PAC}\}.$$

This measures the hardness of the problem. We abbreviate $(0,\delta,K)$-PAC as (δ,K)-PAC, $\mathbb{E}_{\boldsymbol{w}}$ as $\mathbb{E}$, $\mathbb{P}_{\boldsymbol{w}}$ as $\mathbb{P}$, K'_ϵ as K', $\mathcal{T}^\pi$ as $\mathcal{T}$, $\mathbb{T}^*(\boldsymbol{w},\epsilon,\delta,K)$ as $\mathbb{T}^*$ when there is no ambiguity.

3. Algorithm

Our algorithm CascadeBAI(ϵ,δ,K) is presented in Algorithm 1. Intuitively, to identify an ϵ-optimal arm, an agent needs to learn the true weights $w(i)$ of a number of items in $[L]$ by exploring the combinatorial arm space.

At each step t, we classify an item as *surviving*, *accepted* or *rejected*. Initially, all items are surviving and belong to the *survival set* D_t. Over time, an item may be eliminated from D_t, in which case we say that it is *identified*. Once an item is identified, it can be moved to either the *accept set* A_t if it is deemed to be ϵ-optimal, or the *reject set* R_t otherwise. (i) At step 1, all items are in D_1. (ii) At each step t, the agent selects $\min\{K,|D_t|\}$ surviving items with the least number of previous observations, $T_t(i)$'s, pulls them in *ascending* order of the $T_t(i)$, and gets cascading feedback from the user in the form of the $\tilde{k}_t$'s. Similarly to a Racing algorithm (Even-Dar et al., 2002; Maron & Moore, 1994; Heidrich-Meisner & Igel, 2009; Jun et al., 2016), this design of S_t increases the $T_t(i)$'s of all surviving items almost uniformly and avoids the wastage of time steps. (iii) Next, we maintain upper and lower confidence bounds (UCB, LCB) across time to facilitate the identification of items as in Lines 13–17. The confidence radius is defined as follows:

$$C_t(i,\delta) := 4\sqrt{\frac{\log(\log_2[2T_t(i)]/\rho(\delta))}{T_t(i)}},\ \rho(\delta) := \sqrt{\frac{\delta}{12L}}.$$

We set $C_t(i,\delta) = +\infty$ when $T_t(i) = 0$. (iv) Lastly, the algorithm stops once $D_t = \emptyset$, $|A_t| \geq K$ or $|R_t| \geq L-K$.

Algorithm 1 CascadeBAI(ϵ,δ,K)

1: Input: risk δ, tolerance ϵ, size of arm K.
2: Initialize $t=1, D_1=[L], A_1=\emptyset, R_1=\emptyset, T_0(i)=0, \hat{w}_0(i)=0, \forall i$.
3: **while** $D_t \neq \emptyset$, $|A_t|<K$ and $|R_t|<L-K$ **do**
4: Sort the items in D_t according to the number of previous observations: $T_{t-1}(i_1^t) \leq \ldots \leq T_{t-1}(i_{|D_t|}^t)$.
5: **if** $|D_t| \geq K$ **then**
6: pull arm $S_t = (i_1^t, i_2^t, \ldots, i_K^t)$.
7: **else**
8: pull arm $S_t = (i_1^t, i_2^t, \ldots, i_{|D_t|}^t, S'_t)$, where S'_t is any $(K-|D_t|)$-subset of $A_t \bigcup R_t$.
9: **end if**
10: Observe click $\tilde{k}_t \in \{1,\ldots,K,\infty\}$.
11: For each $i \in D_t$, if $W_t(i)$ is observed, set $\hat{w}_t(i) = \frac{T_{t-1}(i)\hat{w}_{t-1}(i)+W_t(i)}{T_{t-1}(i)+1}$, $T_t(i) = T_{t-1}(i)+1$. Otherwise, set $\hat{w}_t(i) = \hat{w}_{t-1}(i), T_t(i) = T_{t-1}(i)$.
12: $k_t = K - |A_t|$.
13: Calculate the UCBs and LCBs for each $i \in D_t$:
$$U_t(i,\delta) = \hat{w}_t(i) + C_t(i,\delta),$$
$$L_t(i,\delta) = \hat{w}_t(i) - C_t(i,\delta).$$
14: Find items in D_t that have the k_t^{th} and $(k_t+1)^{\text{st}}$ largest empirical means:
$j' = \arg\max_{j\in D_t}^{(k_t)} \hat{w}_t(j)$,
$j^* = \arg\max_{j\in D_t}^{(k_t+1)} \hat{w}_t(j)$.
15: $A_{t+1} = A_t \cup \{i \in D_t \mid L_t(i,\delta) > U_t(j^*,\delta) - \epsilon\}$.
16: $R_{t+1} = R_t \cup \{i \in D_t \mid U_t(i,\delta) < L_t(j',\delta) - \epsilon\}$.
17: $D_{t+1} = D_t/(R_{t+1} \bigcup A_{t+1})$.
18: $t = t+1$.
19: **end while**
20: If $|A_t| = K$, output A_t; otherwise, output the first K items that entered A_t.

4. Main Results

We develop an upper bound on the time complexity of CascadeBAI(ϵ,δ,K) and a lower bound on the expected time complexity of any (δ,K)-PAC algorithm. We also discuss the gap between the bounds. We use $c_1, c_2, \ldots$ to denote finite and positive universal constants whose values may vary from line to line. The proofs are sketched in Section 5 and more details are provided in Appendix D.

4.1. Upper Bound

The gaps between the click probabilities determine the hardness to identify the items. The gaps are defined as

$$\Delta_i := \begin{cases} w(i) - w(K+1) & 1 \leq i \leq K \\ w(K) - w(i) & K < i \leq L \end{cases},$$

$$\bar{\Delta}_i := \begin{cases} \Delta_i + \epsilon & 1 \le i \le K \\ \Delta_K - \Delta_i + \epsilon & K < i \le K' \\ \Delta_i - \epsilon & K' < i \le L \end{cases}.$$

Here $\bar{\Delta}_i$ is a slight variation of Δ_i that takes into account the ϵ-optimality of items. Moreover, to correctly identify item i with probability at least $1-\delta/2$, it suffices for our algorithm to observe the item's feedback at least

$$\bar{T}_{i,\delta} := 1 + \left\lfloor \frac{216}{\bar{\Delta}_i^2} \log\left(\frac{2}{\rho(\delta)} \log_2\left(\frac{648}{\rho(\delta)\bar{\Delta}_i^2}\right)\right)\right\rfloor = O\left(\bar{\Delta}_i^{-2} \log\left[\frac{L}{\delta} \log\left(\frac{L}{\delta\bar{\Delta}_i^2}\right)\right]\right)$$

times. Similarly to existing works (Even-Dar et al., 2002; Mannor & Tsitsiklis, 2004), we derive the upper bound with $\bar{\Delta}_i$'s and $\bar{T}_{i,\delta}$'s. A larger $\bar{\Delta}_i$ leads to a smaller $\bar{T}_{i,\delta}$, implying that it requires fewer observations to identify item i correctly. The permutation σ defines the ordering of $\bar{\Delta}$: $\bar{\Delta}_{\sigma(1)} \ge \ldots \ge \bar{\Delta}_{\sigma(L)}$. At step t, we set $\hat{k}_t$ as the number of surviving items in S_t, and $X_{\hat{k}_t;t}$ as the number of observations of them. Note that $\hat{k}_t$ is an r.v. We lower bound $\mathbb{E}X_{k;t}$ with

$$\mu(k,w) := \sum_{i=1}^{k-1} i\cdot w(i) \prod_{j=1}^{i-1} \bar{w}(j) + k \prod_{j=1}^{k-1} \bar{w}(j) \ge \min\left\{\frac{k}{2}, \frac{1}{2w^*}\right\},$$

and upper bound $\mathbb{E}X_{k;t}^2$ with $v(k,w) := \min\{k, \sqrt{2}/w'\}$. We abbreviate $\bar{T}_{i,\delta}$ as $\bar{T}_i$, $\rho(\delta)$ as ρ, $\mu(k,w)$ as μ_k, $v(k,w)$ as v_k when there is no ambiguity. In anticipation of Theorem 4.1, we define three more notations:

$$K_1 := \max\{\, K' - K, \min\{\lfloor 1/w^* \rfloor, K-1\}\,\},$$
$$K_2 := \max\{K' - K, 1\}, \quad M_k := \frac{K+1-k}{\mu_{K+1-k}} - \frac{K-k}{\mu_{K-k}}.$$

Theorem 4.1. *Assume $K' < 2K-1$. With probability at least $1-\delta$, Algorithm 1 outputs an ϵ-optimal arm after at most $(c_1N_1 + c_2N_2 + c_3N_3)$ steps, where*

$$N_1 = \sum_{k=1}^{K-K_2} \frac{v_{K-k+1}^2}{\mu_{K-k+1}^2} \log\left(\frac{1}{\delta} \sum_{j=1}^{K-K_2} \frac{v_{K-j+1}^2}{\mu_{K-j+1}^2}\right),$$

$$N_2 = \frac{1}{\mu_K} \sum_{i=1}^{L-K} \bar{T}_{\sigma(i)},$$

$$N_3 = \sum_{k=2}^{2K-K'} \frac{K-k+1}{\mu_{K-k+1}} \left[\bar{T}_{\sigma(L-K+k)} - \bar{T}_{\sigma(L-K+k-1)}\right] \tag{4.1}$$

$$= \sum_{k=1}^{K-K_1-1} M_k \bar{T}_{\sigma(L-K+k)} + \left(\frac{K_1+1}{\mu_{K_1+1}} - 2\right) \cdot \bar{T}_{\sigma(L-K_1)} + 2\bar{T}_{\sigma(L-K_2)}. \tag{4.2}$$

When $\epsilon = 0$, $\bar{\Delta}_i = \Delta_i$ for all $i \in [L]$ and $K' = K$. We note that it is a waste to pull identified items. This occurs only when $K' < 2K-1$ (see Lemma 5.9) and this scenario is more complicated to analyze. The scenario $K' \ge 2K-1$ is relatively easier to analyze and the result is deferred to Proposition C.1 (see Appendix C).

Interpretation of the bound. The first term N_1 in the bound is unique to the cascading model, which results from the gap between $X_{\hat{k}_t;t}$ and $\mathbb{E}X_{\hat{k}_t;t}$. We can bound N_1 in terms of the maximum and minimum weights, w^* and w'.

Proposition 4.2. *Assume $0 < w' < w^* \le 1$. We have*

$$N_1 \le \begin{cases} 4K \log\left(\frac{4K}{\delta}\right) & 0 < w^* \le 1/K, \\ \frac{8Kw^{*2}}{w'^2} \log\left(\frac{8Kw^{*2}}{\delta w'^2}\right) & 1/K < w^* \le 1. \end{cases}$$

Next, recall that we say that an item is identified by time t if it is put into A_τ or R_τ for some $\tau \le t$. In the worst-case scenario, the agent identifies items in descending order of $\bar{\Delta}_i$'s. With probability at least $1-\delta$, it costs at most c_2N_2 steps to identify items $\sigma(1), \ldots, \sigma(L-K)$ and c_3N_3 is for identifying the remaining ones. More precisely, after item $\sigma(L-K-k-1)$ is identified, the number of steps required for identifying item $\sigma(L-K-k)$ is $(c_3/\mu_{K-k+1}) \cdot (K-k+1)[\bar{T}_{\sigma(L-K+k)} - \bar{T}_{\sigma(L-K+k-1)}]$; we sum these steps up to obtain (4.1). Since the results in many existing works (Even-Dar et al., 2002; Jun et al., 2016) mainly involve $\bar{T}_i$'s, we show the dependence of N_3 on $\bar{T}_i$'s more concretely in (4.2).

Technique. The crucial analytical challenge to derive our bound, especially to establish μ_k, v_k, N_1, is to quantify the impact of partial feedback that results from the cascading model. Firstly, we bound $\mathbb{E}X_{\hat{k}_t;t}$ by exploiting some properties of the cascading feedback. Next, to bound the gap between $\sum_{t=1}^n X_{\hat{k}_t;t}$ and $\sum_{t=1}^n \mathbb{E}X_{\hat{k}_t;t}$ for some $n \in \mathbb{N}$, we propose a novel class of r.v.'s, known as LSG r.v.'s, provide an estimate of a certain LSG parameter, and utilize a Berstein-type concentration inequality to bound the tail probability of a certain LSG r.v.. Details are in Section 5.1.

To facilitate the remaining discussion in Section 4.1, we specialize our analysis and results henceforth to the case of $\epsilon = 0$, in which $\bar{\Delta}_i = \Delta_i$ and the agent aims to find S^*. The remaining results in Section 4.1 can be directly generalized to the scenario of $\epsilon > 0$ by replacing Δ_i's with $\bar{\Delta}_i$'s.

Comparison to the semi-bandit problem. A related algorithm in the setting of semi-bandit feedback and $\epsilon = 0$ is the BATCHRACING Algorithm, which was proposed by Jun et al. (2016). This algorithm has three paramters k, r and b which respectively represent the number of optimal items, the maximum number of pulls of one item at one step and the size of a pulled arm. When $r = 1$ and $b = k$, we denote it as BATRAC(k). The fact that our algorithm observes between 1 and K items per step motivates a comparison among CASCADEBAI$(0, \delta, K)$, BATRAC(K) and BATRAC(1).

Corollary 4.3. *(i) If all $w(i)$'s are at most $1/K$, with prob-*

ability at least $1-\delta$*, Algorithm 1 outputs* S^* *after at most*

$$O\Big(\frac{1}{K}\sum_{i=1}^{L-K}\bar{T}_{\sigma(i)}+\bar{T}_{\sigma(L-1)}\Big) = O\Big(\frac{1}{K}\sum_{i=1}^{L-K}\Delta_{\sigma(i)}^{-2}\log\Big[\frac{L}{\delta}\log\Big(\frac{L}{\delta\Delta_{\sigma(i)}^2}\Big)\Big] + \Delta_{\sigma(L-1)}^{-2}\log\Big[\frac{L}{\delta}\log\Big(\frac{L}{\delta\Delta_{\sigma(L-1)}^2}\Big)\Big]\Big)$$

steps; (ii) if all $w(i)$*'s are at least* $1/2$*, with probability at least* $1-\delta$*, Algorithm 1 outputs* S^* *after at most*

$$O\Big(\sum_{i=1}^{L-1}\bar{T}_{\sigma(i)}\Big) = O\Big(\sum_{i=1}^{L-1}\Delta_{\sigma(i)}^{-2}\log\Big[\frac{L}{\delta}\log\Big(\frac{L}{\delta\Delta_{\sigma(i)}^2}\Big)\Big]\Big)$$

steps.

The results of Corollary 4.3 are intuitive: (i) if all $w(i)$'s are close to 0 (i.e., at most $1/K$), the bound on the time complexity of CascadeBAI$(0,\delta,K)$ is of the same order as that of BatRac(K); (ii) if all $w(i)$'s are close to 1 (i.e., at least $1/2$), the bound corresponds with that of BatRac(1) (Jun et al., 2016). We further upper bound the expected time complexity of our algorithm (denoted by π_1) in these cases.

Proposition 4.4. *(i) If all* $w(i)$*'s are at most* $1/K$*,*

$$\mathbb{E}\mathcal{T}^{\pi_1} \le c_1\log\Big(\frac{1}{\delta}\Big)\cdot\Big\{\frac{1}{K}\sum_{i=1}^{L-K}\Delta_{\sigma(i)}^{-2}\log\Big[L\log\Big(\frac{L}{\Delta_{\sigma(i)}^2}\Big)\Big] + \Delta_{\sigma(L-1)}^{-2}\log\Big[L\log\Big(\frac{L}{\Delta_{\sigma(L-1)}^2}\Big)\Big]\Big\};$$

(ii) if all $w(i)$*'s are at least* $1/2$*,*

$$\mathbb{E}\mathcal{T}^{\pi_1} \le c_2\sum_{i=1}^{L-1}\Delta_{\sigma(i)}^{-2}\log\Big[L\log\Big(\frac{L}{\Delta_{\sigma(i)}^2}\Big)\Big]\log\Big(\frac{1}{\delta}\Big).$$

According to the definition of $\mathbb{T}^*$ in Section 2, $\mathbb{T}^* \le \mathbb{E}\mathcal{T}^{\pi_1}$ and hence also satisfies the above bounds. Corollary 4.3 and Proposition 4.4 indicate that the high probability upper bound on $\mathcal{T}^{\pi_1}$ and the upper bound on $\mathbb{E}\mathcal{T}^{\pi_1}$ are of the same order in the sense that (i) if all $w(i)$'s are at most $1/K$, both upper bounds are $\tilde{O}\big((1/K)\cdot\sum_{i=1}^{L-K}\Delta_{\sigma(i)}^{-2}+\Delta_{\sigma(L-1)}^{-2}\big)$; (ii) if all $w(i)$'s are at least $1/2$, both are $\tilde{O}\big(\sum_{i=1}^{L-1}\Delta_{\sigma(i)}^{-2}\big)$.

Specialization to the case of two click probabilities. We consider a simplified scenario with the following assumption; this allows us to present the upper bound on the time complexity with greater clarity.

Assumption 4.5. *With* $0<w'<w^*\le 1$*, the* K *optimal and* $L-K$ *suboptimal items have click probabilities* w^* *and* w' *respectively.*

Proposition 4.6. *Under Assumption 4.5, (i) if* $0<w^*\le 1/K$*, with probability at least* $1-\delta$*, Algorithm 1 outputs* S^* *after at most*

$$O\Big(\frac{L}{K(w^*-w')^2}\log\Big[\frac{L}{\delta}\log\Big(\frac{L}{\delta(w^*-w')^2}\Big)\Big]\Big)$$

steps; (ii) if $1/K<w^*\le 1$*, with probability at least* $1-\delta$*, Algorithm 1 outputs* S^* *after at most*

$$O\Big(\frac{w^*L}{(w^*-w')^2}\log\Big[\frac{L}{\delta}\log\Big(\frac{L}{\delta(w^*-w')^2}\Big)\Big]+\frac{w^{*2}}{w'^2}\log\Big(\frac{1}{\delta}\Big)\Big)$$

steps.

In the second case, if $L\ge w^*(w^*-w')^2/w'^2$, the first term dominates the bound. For instance, $w'\ge 1/\sqrt{L}$ satisfies this condition.

Proposition 4.7. *Under Assumption 4.5, (i) if* $0<w^*\le 1/K$*,*

$$\mathbb{E}\mathcal{T}^{\pi_1} \le \frac{c_1 L}{K(w^*-w')^2}\log\Big[L\log\Big(\frac{L}{(w^*-w')^2}\Big)\Big]\log\Big(\frac{1}{\delta}\Big);$$

(ii) if $w'\ge 1/2$ *or* $w^*/w'\le 2$*,*

$$\mathbb{E}\mathcal{T}^{\pi_1} \le \frac{c_2 w^* L}{(w^*-w')^2}\log\Big[L\log\Big(\frac{L}{(w^*-w')^2}\Big)\Big]\log\Big(\frac{1}{\delta}\Big).$$

Proposition 4.7 also upper bounds $\mathbb{T}^*$ since $\mathbb{T}^*\le\mathbb{E}\mathcal{T}^{\pi_1}$. It, together with Proposition 4.6 implies that the high probability bound on $\mathcal{T}^{\pi_1}$ and the bound on $\mathbb{E}\mathcal{T}^{\pi_1}$ are of the same order in these cases.

4.2. Lower Bound

We set $\epsilon=0$, in which scenario the agent aims to find an optimal arm S^*. We also upper bound the expected number of observations of items per step by $\tilde{\mu}(K,w)$ where

$$\tilde{\mu}(k,w) = \sum_{i=1}^{k-1} i\cdot w(L+1-i)\prod_{j=1}^{i-1}\bar{w}(L+1-j)+k\prod_{j=1}^{k-1}\bar{w}(L+1-j) \le \min\{1/w', k\}.$$

We write $\tilde{\mu}(k,w)=\tilde{\mu}_k, d(m,n)=\mathrm{KL}(w(m),w(n))$, where $\mathrm{KL}(p,q)=p\log\frac{p}{q}+(1-p)\log\frac{1-p}{1-q}$.

Theorem 4.8. *We have*

$$\mathbb{T}^*\ge\frac{\log(1/2.4\delta)}{\tilde{\mu}_K}\cdot\Big[\sum_{i=1}^{K}\frac{1}{d(i,K+1)}+\sum_{j=K+1}^{L}\frac{1}{d(j,K)}\Big].$$

Comparison to the semi-bandit problem. First, if w' is close to 1 (i.e., $w'\ge 1/2$), $\tilde{\mu}_K=1/w'\le 2$, i.e., at one step, the agent observes at most 2 items in expectation. We can recover the lower bound in Kaufmann et al. (2016) by replacing $\tilde{\mu}_K$ with 1, which is of the same order as our bound in this case. Next, if w' is close to 0 (i.e., $w'\le 1/K$), the agent observes $\tilde{\mu}_K=K$ items in expectation. Then the bound is the same as that incurred by pulling K items per step and getting semi-bandit feedback, which is

$$\frac{\log(1/2.4\delta)}{K}\cdot\Big[\sum_{i=1}^{K}\frac{1}{d(i,K+1)}+\sum_{j=K+1}^{L}\frac{1}{d(j,K)}\Big].$$

Specialization to the case of two click probabilities.

Corollary 4.9. *Under Assumption 4.5, we have*

$$\mathbb{T}^* \geq \frac{\mathrm{KL}(1-\delta,\delta)}{\tilde{\mu}_K} \cdot \left[\frac{K}{\mathrm{KL}(w^*,w')} + \frac{L-K}{\mathrm{KL}(w',w^*)}\right] = \Omega\Big(\min\{w', 1-w^*\} \cdot \frac{Lw'}{(w^*-w')^2} \log\Big[\frac{1}{\delta}\Big]\Big).$$

where $\tilde{\mu}_K = [1-(1-w')^K]/w' \leq 1/w'$.

4.3. Comparison of the Upper and Lower Bounds

To see whether the upper and lower bounds on $\mathbb{T}^*$ match, we set $\epsilon = 0$ and consider the following simplified cases.

Corollary 4.10. *Set* $\epsilon = 0$. *(i) If* $0 < w^* \leq 1/K$,

$$\mathbb{T}^* \in \tilde{\Omega}\Big(\frac{Lw'^2}{(w^*-w')^2}\Big) \bigcap \tilde{O}\Big(\frac{L}{K(w^*-w')^2}\Big);$$

(ii) if $1 > w' \geq 1/2$,

$$\mathbb{T}^* \in \tilde{\Omega}\Big(\frac{Lw'(1-w^*)}{(w^*-w')^2}\Big) \bigcap \tilde{O}\Big(\frac{w^*L}{(w^*-w')^2}\Big).$$

The upper bounds above are achieved by Algorithm 1.

In the first case, the gap between the upper and lower bounds is manifested in the terms $1/K$ and w'^2. In the second case, the gap is manifested in w^* and $w'(1-w^*)$.

5. Proof Sketch

5.1. Analysis of Partial Feedback for Cascading Bandits

At a high level, the time complexity $\mathcal{T}$ can be established by analyzing $\sum_{t=1}^{\mathcal{T}} X_{\hat{k}_t;t}$ and $X_{\hat{k}_t;t}$. The first term is determined by $\bar{T}_{i,\delta}$'s, the number of observations that guarantees the correct identification of items with high probability. These $\bar{T}_{i,\delta}$'s are invariant to the scenario whether the agent receives semi-bandit or partial feedback from the user. The second term $X_{\hat{k}_t;t}$ equals to $\hat{k}_t$ in the semi-bandit feedback setting while it is an r.v. in the partial feedback setting. Since $\bar{T}_{i,\delta}$'s have already been studied by a number of works on the semi-bandit feedback (Even-Dar et al., 2002; Jun et al., 2016), the crucial challenge of analyzing cascading bandits is to estimate $X_{\hat{k}_t;t}$ probabilitistically.

According to Algorithm 1, $\hat{k}_t = \min\{K, |D_t|\}$. When $\hat{k}_t = K \leq |D_t|$, the agent pulls K surviving (i.e., not identified) items. Otherwise, the agent pulls all surviving items first and then complements S_t with some identified items. In the cascading bandit setting, the agent observes only one item when the first item i_1^t is clicked, and the corresponding probability is $w(i_1^t)$; the agent observes two items when i_1^t is not clicked but i_2^t is clicked, and the probability is $[1-w(i_1^t)]w(i_2^t)$; and so on. Therefore,

$$\mathbb{E}X_{\hat{k}_t;t} = \sum_{i=1}^{\hat{k}_t-1} i \cdot \left[\prod_{j=1}^{i-1} \bar{w}(i_j^t) w(i_i^t)\right] + \hat{k}_t \prod_{j=1}^{\hat{k}_t-1} \bar{w}(i_j^t).$$

Since $\mathbb{E}X_{\hat{k}_t;t}$ depends only on S_t (the pulled arm at step t) and S_t is learnt online, it is difficult to estimate $\mathbb{E}X_{\hat{k}_t;t}$ for each step separately. Therefore, the second best thing one can do is to bound $\mathbb{E}X_{\hat{k}_t;t}$ as a function of $\hat{k}_t$ and $\boldsymbol{w}$. We now present some properties of $\mathbb{E}X_{\hat{k}_t;t}$.

Theorem 5.1. *Consider a set of items with weights* $\boldsymbol{u} = (u_1, \ldots, u_k)$ *such that* $u_1 \geq \ldots \geq u_k$, *and let* $\mu_k(\boldsymbol{u}, I)$ *be the expected number of observations when items are placed with order* I. *Let* $I_{dec} = (1, \ldots, k)$, $I_{inc} = (k, \ldots, 1)$, *and* I *be any order, then*
(i) boundedness: $\mu_k(\boldsymbol{u}, I_{dec}) \leq \mu_k(\boldsymbol{u}, I) \leq \mu_k(\boldsymbol{u}, I_{inc})$;
(ii) monotonicity: let $\boldsymbol{v} = (v_1, \ldots, v_k)$ *be another vector of weights, then* $\mu_k(\boldsymbol{u}, I) \geq \mu_k(\boldsymbol{v}, I)$ *if* $u_i \leq v_i$ *for all* $i \in [k]$.

Theorem 5.1 implies that when $\boldsymbol{w}$ is fixed, $\mathbb{E}X_{k;t}$ attains its minimum when the agent pulls items $1, 2, \ldots, k$ in this order and attains its maximum when the agent pulls $L, L-1, \ldots, L-K+1$ in this order. Moreover, if $w(i) = w^*$ for all $i \in [k]$, $\mathbb{E}X_{k;t}$ is even smaller; if $w(j) = w'$ for all $j \in \{L-k+1, \ldots, L\}$, $\mathbb{E}X_{k;t}$ is even larger. This observation inspires Lemma 5.2.

Lemma 5.2. *For any* k, t,

$$\min\Big\{\frac{k}{2}, \frac{1}{2w^*}\Big\} \leq \mu_k \leq \mathbb{E}X_{k;t} \leq \tilde{\mu}_k \leq \min\Big\{\frac{1}{w'}, k\Big\}.$$

Next, since $X_{k;t}$, instead of $\mathbb{E}X_{k;t}$, affects the dynamics, we examine the gap between $\sum_{t=1}^n X_{k;t}$ and $\sum_{t=1}^n \mathbb{E}X_{k;t}$. Clearly, a tight concentration inequality is essential to estimate this gap well. Since $X_{k;t}$ is a bounded r.v., there are some applicable Bernstein-type inequalities. For instance, we can apply Azuma's inequality to analyze SG r.v.'s. However, (i) it is challenging to find an SG parameter of $X_{k;t}$ that is good enough for our purpose (a more detailed explanation is provided after Lemma 5.6), and (ii) we only require a one-sided concentration inequality. Hence, we resort to defining a new class of r.v.'s — known as LSG r.v.'s — and provide an estimate of the relevant LSG parameter.

Definition 5.3 (LSG). *An r.v.* X *is* v*-LSG* ($v \geq 0$) *if*

$$\mathbb{E}[\exp[\lambda(X - \mathbb{E}X)]\,] \leq \exp(v^2\lambda^2/2), \quad \forall \lambda \leq 0.$$

Theorem 5.4. *Let* X *be an almost surely bounded nonnegative r.v.. If* $\mathbb{E}X^2 \leq v^2$, *then* X *is* v*-LSG.*

Furthermore, we bound $\mathbb{E}X_{k;t}^2$ (Lemma 5.5) and adapt a variation of Azuma's inequality as in Theorem B.1 (Shamir, 2011) to evaluate the dependence between the number of observations and the number of time steps.

Lemma 5.5. *For any* k, t, $\mathbb{E}X_{k;t}^2 \leq v_k^2 = \min\{k^2, 2/w'^2\}$.

Lemma 5.6. *For any $k, t, \delta > 0$, set*

$$\mathcal{E}^* := \left\{ \sum_{t=1}^{n} X_{k;t} \le n\mu_k - \sqrt{2nv_k^2 \log\left(\frac{1}{\delta}\right)} \right\},$$

then $\Pr(\mathcal{E}^*) \le \delta$. Further when $\overline{\mathcal{E}^*}$ holds, for any $T > 0$, $\sum_{t=1}^{n} X_{k;t} \le T$ implies that $n \le 2T/\mu_k + 2\log(1/\delta)v_k^2/\mu_k^2$.

Lemma 5.6 implies that with high probability, we can lower bound the amount of observations on the surviving items over the whole horizon. Subsequently, with probability at least $1-\delta$, the agent would have received sufficiently many observations on the surviving items to return an ϵ-optimal arm after at most $(c_1N_1 + c_2N_2 + c_3N_3)$ time steps (see Theorem 4.1). The lemma also indicates that a smaller LSG/SG parameter of $X_{k;t}$ leads to a smaller upper bound on the number of time steps. Since we can show $X_{k;t}$ is v_k-LSG but cannot show it is v_k-SG (a detailed discussion is deferred to Appendix D.9), it is beneficial to consider the class of LSG distributions for our problem. The class of LSG r.v.'s and the general estimate of the LSG parameter, which is crucial for the utilization of the concentration inequality, may be of independent interest.

5.2. Proof Sketch of Theorem 4.1

Concentration. As the algorithm proceeds, the agent moves items from D_t to A_t or R_t according to the confidence bounds of all surviving items in D_t. This motivates us to define a "nice event"

$$\mathcal{E}(i,\delta) := \{\forall t \ge 1, L_t(i,\delta) \le \hat{w}_t(i) \le U_t(i,\delta)\}.$$

To show that $\bigcap_{i=1}^{L} \mathcal{E}(i,\delta)$ holds with high probability, we utilize Theorem B.2 (Jamieson et al., 2014; Jun et al., 2016) and the SG property of $W_t(i)$ (the r.v. that reflects whether item i is clicked at time step t).

Lemma 5.7. *For any $\delta \in [0,1]$, $\mathbb{P}\big(\bigcap_{i=1}^{L} \mathcal{E}(i,\delta)\big) \ge 1 - \delta/2$.*

Sufficient observations. Next, we assume $\bigcap_{i=1}^{L} \mathcal{E}(i,\delta)$ holds and find the number of observations that guarantees the correct identification of an item. To facilitate the analysis of the expected time complexity (Proposition 4.4, 4.7), we assume $\bigcap_{i=1}^{L} \mathcal{E}(i,\delta')$ holds for a fixed $\delta' \in (0,\delta]$ in Lemma 5.8, which generalizes Jun et al. (2016, Lemma 2).

Lemma 5.8. *Fix any $0 < \delta' \le \delta$, assume $\bigcap_{i=1}^{L} \mathcal{E}(i,\delta')$ holds. Set $T_t' := \min_{i \in D_t} T_t(i)$, then for any time step t,*

$$\forall i \le K', T'(t) \ge \bar{T}_{i,\delta'} \Rightarrow L_t(i,\delta) > U_t(j^*,\delta) - \epsilon \Rightarrow i \in A_t,$$
$$\forall i > K', T'(t) \ge \bar{T}_{i,\delta'} \Rightarrow U_t(i,\delta) < L_t(j',\delta) - \epsilon \Rightarrow i \in R_t.$$

Lemmas 5.7 and 5.8 imply that with sufficiently many observations, the agent can correctly identify items with probability at least $1-\delta/2$.

Time complexity. Subsequently, we observe that our algorithm stops before identifying all items.

Lemma 5.9. *Assume $\bigcap_{i=1}^{L} \mathcal{E}(i,\delta)$ holds. Algorithm 1 stops after identifying at most $L - \max\{K' - K, 1\}$ items.*

Lemma 5.9 indicates that it suffices to count the number of time steps needed to identify at most $L - K' + K$ items.

We consider the worst case in which the agent identifies items in descending order of the $\bar{\Delta}_i$'s. We divide the whole horizon into several phrases according to $|D_t|$, the number of surviving items. During each phrase, we upper bound the required number of observations with Lemma 5.8; then Lemma 5.6 helps to upper bound the required number of time steps with high probability. Lastly, we bound the total error probability by $\delta/2$ and utilize the Lagrange multipliers to solve the following problem:

$$\max_{\delta_k : 1 \le k \le 2K-K'} \sum_{k=1}^{2K-K'} \frac{2v_{K-k+1}^2}{\mu_{K-k+1}^2} \log \delta_k \ \text{ s.t. } \sum_{k=1}^{2K-K'} \delta_k \le \delta/2.$$

Altogether, we upper bound the time complexity.

5.3. Proof Sketch of Theorem 4.8

Construct instances. To begin, we fix $\alpha > 0$ and define a class of $L+1$ instances, indexed by $\ell = 0, 1, \ldots, L$:
- under instance 0, we have $\{w(1), w(2), \ldots, w(L)\}$,
- under instance ℓ, we have $\{w(1), w(2), \ldots, w(\ell-1), w^{(\ell)}(\ell), w(\ell+1), \ldots, w(L)\}$;

where we define $w_\ell^{(\ell)}$'s so that they satisfy

$$\begin{aligned}
1 \le i \le K:\ & w^{(i)}(i) < w(K+1), \\
& \mathrm{KL}(w(i), w(K+1)) < \mathrm{KL}(w(i), w^{(i)}(i)), \\
& \mathrm{KL}(w(i), w^{(i)}(i)) < \mathrm{KL}(w(i), w(K+1)) + \alpha, \\
K < j \le L:\ & w^{(j)}(j) > w(K), \\
& \mathrm{KL}(w(j), w(K)) < \mathrm{KL}(w(j), w^{(j)}(j)), \\
& \mathrm{KL}(w(j), w^{(j)}(j)) < \mathrm{KL}(w(j), w(K)) + \alpha.
\end{aligned}$$

In particular, $S^* \in [K]^{(K)}$ is optimal under instance 0, while suboptimal under instance $1 \le \ell \le L$. Bearing the risk of overloading the notations, under instance ℓ, we denote $S^{*,\ell}$ as an optimal arm, $S_t^{\pi,\ell}$ as the arm chosen by algorithm π at step t and $\boldsymbol{O}_t^{\pi,\ell}$ as the corresponding stochastic outcome (see its definition in Section 2).

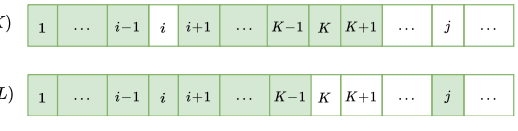


Figure 5.1: Optimal set $S^{*,\ell}$ in instance ℓ (shaded in green)

KL divergence. We first apply chain rule to decompose $\mathrm{KL}\big(\{S_t^{\pi,0}, \boldsymbol{O}_t^{\pi,0}\}_{t=1}^{\mathcal{T}}, \{S_t^{\pi,\ell}, \boldsymbol{O}_t^{\pi,\ell}\}_{t=1}^{\mathcal{T}}\big)$.

Lemma 5.10 (Cheung et al. (2018, Lemma 6.4))**.** *For any*

$1 \le \ell \le L$,

$$\mathrm{KL}\big(\{S_t^{\pi,0}, \boldsymbol{O}_t^{\pi,0}\}_{t=1}^{\mathcal{T}}, \{S_t^{\pi,\ell}, \boldsymbol{O}_t^{\pi,\ell}\}_{t=1}^{\mathcal{T}}\big)$$
$$= \sum_{t=1}^{\mathcal{T}} \sum_{s_t \in [L]^{(K)}} \Pr[S_t^{\pi,0} = s_t] \mathrm{KL}\big(P_{\boldsymbol{O}_t^{\pi,0}|S_t^{\pi,0}}(\cdot \mid s_t) \big\| P_{\boldsymbol{O}_t^{\pi,\ell}|S_t^{\pi,\ell}}(\cdot \mid s_t)\big).$$

Next, we lower bound $\mathbb{E}[T_{\mathcal{T}}(\ell)]$ with the KL divergence by applying a result from Kaufmann et al. (2016).

Lemma 5.11. *For any* $1 \le \ell \le L$,

$$\mathrm{KL}\big(\{S_t^{\pi,0}, \boldsymbol{O}_t^{\pi,0}\}_{t=1}^{\mathcal{T}}, \{S_t^{\pi,\ell}, \boldsymbol{O}_t^{\pi,\ell}\}_{t=1}^{\mathcal{T}}\big)$$
$$= \mathbb{E}[T_{\mathcal{T}}(\ell)] \cdot \mathrm{KL}\big(w(\ell), w^{(\ell)}(\ell)\big) \ge \sup_{\mathcal{E} \in \mathcal{T}} \mathrm{KL}\big(\mathbb{P}_0(\mathcal{E}), \mathbb{P}_\ell(\mathcal{E})\big).$$

Define the event $\mathcal{E} := \{\hat{S}^\pi \in [K]^{(K)}\} \in \mathcal{F}_{\mathcal{T}}$. We establish that, for any (δ, K)-PAC algorithm, $\mathbb{P}_0(\mathcal{E}) \ge 1 - \delta$ and $\mathbb{P}_\ell(\mathcal{E}) \le \delta$ $(\forall 1 \le \ell \le L)$. Lastly, by revisiting the definition of $X_{k;t}$ in Section 4.1, we see that $\tilde{\mu}_K$ also upper bounds the expected number of observations of items at one step for any (δ, K)-PAC algorithm (Lemma 5.2). This allows us to lower bound $\mathbb{T}^*$.

6. Experiments

We evaluate the performance of CascadeBAI(ϵ, δ, K) and some related algorithms. For each choice of algorithm and instance, we run 20 independent trials. The standard deviations of the time complexities of our algorithm are negligible compared to the averages, and therefore are omitted from the plots. More details are provided in Appendix E.

6.1. Order of Pulled Items

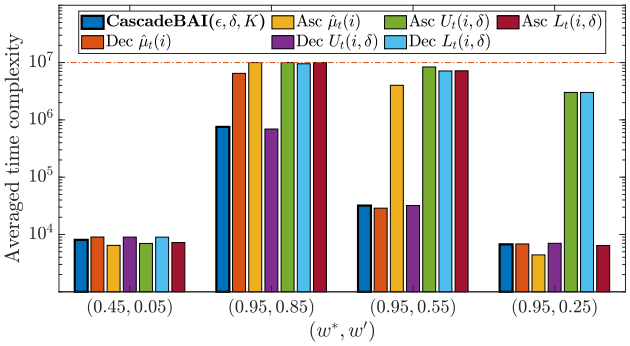
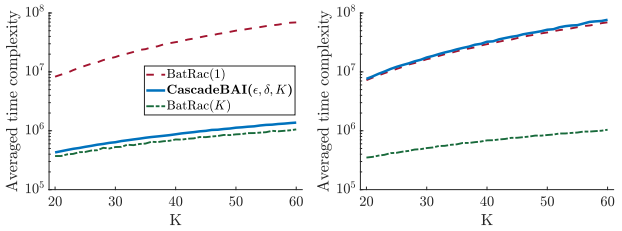


Figure 6.1: Average time complexity incurred by different sorting order of S_t: ascending order of $T_i(t)$ (Algorithm 1), ascending/descending order of $\hat{\mu}_t(i)/U_t(i)/L_t(i)$ with $L = 64$, $K = 16$, $\delta = 0.1$ and $\epsilon = 0$ in the cascading bandits.

As shown in Lines 5–9 of Algorithm 1, CascadeBAI(ϵ, δ, K) sorts items in S_t based on *ascending* order of $T_{t-1}(i)$'s. This order is crucial for proving our theoretical results. To learn the impact of ordering on the time complexity, we evaluate the empirical performance of sorting S_t in descending or ascending order of $\hat{w}_t(i)$'s, $U_t(i, \delta)$'s or $L_t(i, \delta)$'s. We compare these methods under various problem settings and set the maximum time step as 10^7. Figure 6.1 shows that CascadeBAI(ϵ, δ, K) empirically performs as well as the best among the other heuristics, but CascadeBAI(ϵ, δ, K) is the only one with a theoretical guarantee.

6.2. Comparison to Semi-feedback Setting

We compare CascadeBAI$(0, \delta, K)$, BatRac(K) and BatRac(1) (Jun et al., 2016) empirically. In Figure 6.2, if w^*, w' are sufficiently small as the parameters shown in subfigure (a), CascadeBAI$(0, \delta, K)$ performs similarly to BatRac(K); if w^*, w' are large as in subfigure (b), it behaves similarly to BatRac(1). This corroborates the implications of Corollary 4.3.



(a) $w^* = \frac{1}{K}, w' = \frac{1}{K^2}$ (b) $w^* = 1 - \frac{1}{K^2}, w' = 1 - \frac{1}{K}$

Figure 6.2: Average time complexity of CascadeBAI(ϵ, δ, K), BatRac(1), BatRac(K) with $L = 128$, $\delta = 0.1$, $\epsilon = 0$, $K = 20, \ldots, 60$.

6.3. Further Empirical Evidence

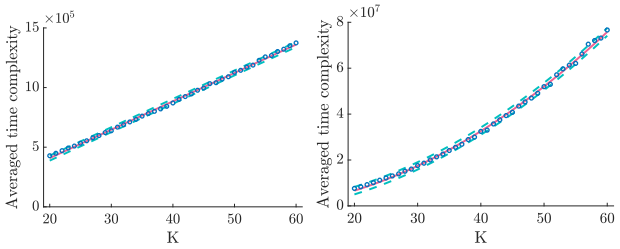
Our analysis of the cascading feedback involves v_k, μ_k in the upper bound of the time complexity; these parameters depend strongly on w^*, w' and K (Lemma 5.2, 5.5). Hence, to assess the tightness of our analyses, we consider several simplified cases by choosing w^*, w' as functions of K and examine whether the dependence of the resultant time complexity (Proposition 4.6) on K is materialized through numerical experiments.

Table 6.1: Upper bounds on the time complexity of Algorithm 1 with $L = 128$, $K = 20, \ldots, 60$, $\delta = 0.1$, $\epsilon = 0$ (Proposition 4.6), and their fitting results.

w^*	w'	Upper bound	Fit. model	R^2-stat.
$1/K$	$1/K^2$	$\tilde{O}(K)$	$c_1 K + c_2$	0.999
$1 - 1/K^2$	$1 - 1/K$	$\tilde{O}(K^2)$	$c_1 K^2 + c_2$	0.999
$1/\sqrt{K}$	$1/K$	$\tilde{O}(K)$	$c_1 K + c_2$	0.973
$1 - 1/K$	$1 - 1/\sqrt{K}$	$\tilde{O}(K)$	$c_1 K + c_2$	1.000
$1 - 1/K$	$1/K$	$\tilde{O}(K^2)$	$c_1 K^2 + c_2$	0.992

We fit a model to the averaged time complexity under each setting as stated in Table 6.1. In each case, the R^2-statistic is almost 1, implying that the variability of the time complex-

ity is almost fully explained by the proposed polynomial model (Glantz et al., 1990). Therefore, the empirical results show that the dependence of the upper bound on K (Proposition 4.6) is rather tight, which implies that using the new concept of LSG r.v.'s, our quantifications of the cascading feedback are also rather tight.



(a) $w^* = \frac{1}{K}, w' = \frac{1}{K^2}$ (b) $w^* = 1 - \frac{1}{K^2}, w' = 1 - \frac{1}{K}$

Figure 6.3: Fit the averaged time complexity with functions of K for two cases. Fix $L = 128$, $\delta = 0.1$, $\epsilon = 0$. Blue dots are the averaged time complexity, red line is the fitted curve, and cyan dashed lines show the 95% confidence interval.

7. Summary and Future Work

This work presents the first theoretical analysis of best arm identification problem with partial feedback. We also show that the upper bound for the CASCADEBAI(ϵ, δ, K) algorithm closely matches the lower bound in some cases. Empirical experiments further support the theoretical analysis. Moreover, the relation between the second moment and the LSG property of a bounded random variable may be of independent mathematical interest.

The assumption of $w^* < 1/K$ (ensuring tightness of the sample complexity in Corollary 4.10) is relevant in practical applications since CTRs are low in real applications. For most applications (e.g. online advertising), K is small ($\approx$ 5-10), so our assumption is reasonable. We are cautiously optimistic that, the framework could be enhanced for better bounds in the remaining less practically relevant regime, which is an avenue for future work.

The following are some more avenues for further investigations. First, we note that estimating the number of observations per time step is key to establishing a high probability bound on the total number of time steps. In this work, we bound the expectation of this term with w^* and w' (Lemma 5.2). This bound is tight in some cases (cf. Corollary 4.10). These include the difficult case where all click probabilities $w(i)$'s are close to one another and hence the gaps are small. Nevertheless, a tighter bound for each individual time step may improve the results; this, however, will require a more elaborate and delicate analysis of the stochastic outcomes and their impacts at each time step. This is especially so since the order of selected items also affects the number of observations in the cascading bandit setting. Secondly, this work focuses on the fixed-confidence setting of the BAI problem. We see that the consideration of the fixed-budget setting for cascading bandits is still not available. It is envisioned that the analysis of the statistical dependence between the number of observations and time steps would be non-trivial. Thirdly, we envision that the analysis may be generalized to the contextual setting (Soare et al., 2014; Tao et al., 2018; Degenne et al., 2020).

Acknowledgment

We thank the reviewers for their insightful comments. We also thank Ms. Yuko Kuroki for her comments on the manuscript and for pointing us to pertinent references. This work is partially funded by a National University of Singapore Start-Up Grant (R-266-000-136-133) and a Singapore National Research Foundation (NRF) Fellowship (R-263-000-D02-281).

References

Agarwal, A., Agarwal, S., Assadi, S., and Khanna, S. Learning with limited rounds of adaptivity: Coin tossing, multi-armed bandits, and ranking from pairwise comparisons. In Kale, S. and Shamir, O. (eds.), *Proceedings of the 2017 Conference on Learning Theory*, volume 65 of *Proceedings of Machine Learning Research*, pp. 39–75, Amsterdam, Netherlands, 07–10 Jul 2017. PMLR.

Anantharam, V., Varaiya, P., and Walrand, J. Asymptotically efficient allocation rules for the multiarmed bandit problem with multiple plays–Part I: I.I.D. rewards. *IEEE Transactions on Automatic Control*, 32(11):968–976, November 1987.

Audibert, J.-Y. and Bubeck, S. Best arm identification in multi-armed bandits. In *COLT-23th Conference on learning theory-2010*, pp. 13–p, 2010.

Auer, P., Cesa-Bianchi, N., and Fischer, P. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3):235–256, 2002.

Carpentier, A. and Locatelli, A. Tight (lower) bounds for the fixed budget best arm identification bandit problem. In *Conference on Learning Theory*, pp. 590–604, 2016.

Chen, S., Lin, T., King, I., Lyu, M. R., and Chen, W. Combinatorial pure exploration of multi-armed bandits. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q. (eds.), *Advances in Neural Information Processing Systems 27*, pp. 379–387. Curran Associates, Inc., 2014.

Cheung, W. C., Tan, V. Y. F., and Zhong, Z. Thompson sampling for cascading bandits. In *arXiv: 1810.01187*, 2018. URL http://arxiv.org/abs/1810.01187.

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

Cover, T. M. and Thomas, J. A. *Elements of information theory*. John Wiley & Sons, 2012.

Craswell, N., Zoeter, O., Taylor, M., and Ramsey, B. An experimental comparison of click position-bias models. In *Proceedings of the 1st ACM International Conference on Web Search and Data Mining*, pp. 87–94, 2008.

Degenne, R., Ménard, P., Shang, X., and Valko, M. Gamification of pure exploration for linear bandits. In *arXiv:2007.00953*, 2020. URL http://arxiv.org/abs/2007.00953.

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

Glantz, S. A., Slinker, B. K., and Neilands, T. B. *Primer of applied regression and analysis of variance*, volume 309. McGraw-Hill New York, 1990.

Heidrich-Meisner, V. and Igel, C. Hoeffding and bernstein races for selecting policies in evolutionary direct policy search. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 401–408. ACM, 2009.

Jamieson, K. and Nowak, R. Best-arm identification algorithms for multi-armed bandits in the fixed confidence setting. In *2014 48th Annual Conference on Information Sciences and Systems (CISS)*, pp. 1–6. IEEE, 2014.

Jamieson, K., Malloy, M., Nowak, R., and Bubeck, S. lil'ucb: An optimal exploration algorithm for multi-armed bandits. In *Conference on Learning Theory*, pp. 423–439, 2014.

Jun, K.-S., Jamieson, K. G., Nowak, R. D., and Zhu, X. Top arm identification in multi-armed bandits with batch arm pulls. In *AISTATS*, pp. 139–148, 2016.

Kalyanakrishnan, S., Tewari, A., Auer, P., and Stone, P. Pac subset selection in stochastic multi-armed bandits. In *ICML*, volume 12, pp. 655–662, 2012.

Kaufmann, E., Cappé, O., and Garivier, A. On the complexity of best-arm identification in multi-armed bandit models. *The Journal of Machine Learning Research*, 17 (1):1–42, 2016.

Kuroki, Y., Xu, L., Miyauchi, A., Honda, J., and Sugiyama, M. Polynomial-time algorithms for multiple-arm identification with full-bandit feedback. Accepted by *Neural Computataion*. 2019. URL http://arxiv.org/abs/1902.10582.

Kveton, B., Wen, Z., Ashkan, A., Eydgahi, H., and Eriksson, B. Matroid bandits: Fast combinatorial optimization with learning. In *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence*, UAI'14, pp. 420–429, 2014.

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

Kveton, B., Wen, Z., Ashkan, A., and Szepesvári, C. Combinatorial cascading bandits. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, pp. 1450–1458, 2015b.

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

Li, L., Chu, W., Langford, J., and Schapire, R. E. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th International Conference on World Wide Web*, pp. 661–670, 2010.

Li, S., Wang, B., Zhang, S., and Chen, W. Contextual combinatorial cascading bandits. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48, pp. 1245–1253, 2016.

Mannor, S. and Tsitsiklis, J. N. The sample complexity of exploration in the multi-armed bandit problem. *Journal of Machine Learning Research*, 5(Jun):623–648, 2004.

Maron, O. and Moore, A. W. Hoeffding races: Accelerating model selection search for classification and function approximation. In *Advances in neural information processing systems*, pp. 59–66, 1994.

Qin, L., Chen, S., and Zhu, X. Contextual combinatorial bandit and its application on diversified online recommendation. In *SDM*, pp. 461–469. SIAM, 2014.

Rejwan, I. and Mansour, Y. Combinatorial bandits with full-bandit feedback: Sample complexity and regret minimization. In *arXiv:1905.12624*, 2019. URL http://arXiv.org/abs/1905.12624.

Rejwan, I. and Mansour, Y. Top-k combinatorial bandits with full-bandit feedback. In Kontorovich, A. and Neu, G. (eds.), *Proceedings of the 31st International Conference*

on Algorithmic Learning Theory, volume 117 of *Proceedings of Machine Learning Research*, pp. 752–776, San Diego, California, USA, 08 Feb–11 Feb 2020. PMLR.

Shamir, O. A variant of azuma's inequality for martingales with subgaussian tails. *arXiv preprint arXiv:1110.2392*, 2011.

Soare, M., Lazaric, A., and Munos, R. Best-arm identification in linear bandits. In *Advances in Neural Information Processing Systems*, pp. 828–836, 2014.

Tao, C., Blanco, S., and Zhou, Y. Best arm identification in linear bandits with linear dimension dependency. In Dy, J. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 4877–4886, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR.

Wang, Q. and Chen, W. Improving regret bounds for combinatorial semi-bandits with probabilistically triggered arms and its applications. In *Advances in Neural Information Processing Systems*, pp. 1161–1171, 2017.

Zong, S., Ni, H., Sung, K., Ke, N. R., Wen, Z., and Kveton, B. Cascading bandits for large-scale recommendation problems. In *Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence*, UAI'16, pp. 835–844, 2016.