Exploration, Exploitation, and Engagement in Multi-Armed Bandits with Abandonment

Zixian Yang
EECS
University of Michigan
Ann Arbor, MI, USA
zixian@umich.edu

Xin Liu
School of Information Science and Technology
ShanghaiTech University
Shanghai, China
liuxin7@shanghaitech.edu.cn

Lei Ying
EECS
University of Michigan
Ann Arbor, MI, USA
leiying@umich.edu

Recommendation algorithms have become increasingly important in many online platforms such as online education, TikTok, YouTube Shorts, advertising platforms, etc. Multi-armed bandit (MAB) [2] is a classic problem which can model these recommendation systems. Each arm in MAB corresponds to a specific type of item in the recommendation system. The recommendation of an item of the ith type is regarded as a pull of arm a_i. Taking recommending short videos as an example, each arm a_i represents a class of similar videos (e.g. videos from the same dancer). For simplicity, we assume the reward is 1 if the user likes the recommended item and is 0 otherwise. In a traditional MAB problem, the learner can continue to play the arms with the goal of maximizing the average reward, which either assumes a single user stays in the system for a long period of time or assumes the learner is recommending a single item to each user with a large number of users. While this traditional MAB formulation models recommendation systems such as online advertising well, there are new recommendation systems that are significantly different from these traditional models. In these new recommendation systems, such as TikTok or ALEKS, the learner continuously recommends videos/contents to a user, and the user, other than like or dislike the item, may abandon the system if the recommended items cannot engage the user, and come back later. For example, a user watches TikTok or YouTube Shorts for some period of time, where the duration depends on how interesting/engaging the videos are, then leaves the systems, and comes back later.

This makes the problem different from traditional MAB because the objective now is to maximize the total reward per episode (visit) instead of the average reward per pull. Therefore, in addition to finding the most rewarding arm, the learner also needs to continue to engage the user to maximize the number of plays of each episode. Because of the abandonment, the exploration needs to be carefully designed so that the learner should explore (recommend new types of items) when the user is less likely to abandon the system. In other words, we need to consider exploration-exploitation-engagement in this problem. We denote this problem by MAB-A. In MAB-A, the system recommends one item at a time to the user. The user may or may not like the item, and they may abandon the system with a certain probability (called abandonment probability in this paper) based on current and previous experience. The objective is to maximize the total reward per episode, where an episode ends when the user abandons (leaves) the system temporarily.

This paper is an extended abstract of [1]. The work of Zixian Yang and Lei Ying is supported in part by NSF under grants 2002608 and 2001687.

I. Model and Preliminaries

The MAB-A problem is defined as follows. Let M ($M \geqslant 2$) be the number of arms and denote the set of arms by $\{a_1, a_2, \cdots, a_M\}$. Assume that the rewards of pulling the arms are i.i.d. Bernoulli random variables with unknown mean $\mu(a_i)$, $i \in \{1, 2, ..., M\}$. Consider K episodes in total, where each episode represents a single visit of a user and an episode ends when the user abandons the system temporarily. The process of the kth episode goes as follows. At step $h = 1$, an initial state $S_{k,1} \in \{0, 1\}$ is sampled from an arbitrary distribution. At step $h = 2, 3, ...$, the state is defined by $S_{k,h} := R_{k,h-1}$, where $R_{k,h-1}$ is the reward obtained at the previous step $h-1$. Then an arm $A_{k,h} \in \{a_1, ..., a_M\}$ is pulled and a Bernoulli random reward $R_{k,h} \in \{0, 1\}$ is obtained with mean $\mu(A_{k,h})$. Given $(S_{k,h}, R_{k,h})$, abandonment occurs with probability $q(S_{k,h}, R_{k,h})$. If the abandonment occurs, the terminal state g is reached, i.e., $S_{k,h+1} = g$, which terminates the current episode k. Otherwise, the process goes to the next step. Therefore, the process of one episode is an MDP with state space $\mathcal{S} = \{0, 1, g\}$, action space $\mathcal{A} = \{a_1, ..., a_M\}$, and Bernoulli random rewards. The state can be interpreted as the experience of the user. At the first step ($h = 1$) in each episode, the initial state $S_{k,1}$ can be interpreted as the user's first impression. We make the following assumption.

Assumption 1 *Assume* $q(i, j) \geqslant q(i', j')$ *if* $i + j < i' + j'$, $q(0, 0) > 0$ *and* $0 < \mu(a_M) \leqslant ... \leqslant \mu(a_2) < \mu(a_1) < 1$.

The assumption on $q(\cdot, \cdot)$ implies the abandonment probability becomes larger when the user's experience becomes worse. The assumptions $q(0, 0) > 0$ and $\mu(a_i) < 1, \forall i$ ensures that all policies are proper. That is, all policies lead to the terminal state g with probability one, regardless of the initial state [3]. We next define the baseline, i.e. the reward under a genie-aided

(model-based) optimal policy π^*, which knows the model perfectly. Then we have the following:

Lemma 1 *Under Assumption 1, π^* is always pulling arm a_1.*

Let $\pi : \mathcal{S} \times \Phi \to \mathcal{A}$ denote a deterministic policy such that $A_{k,h} = \pi(S_{k,h}, \phi_{k,h})$, where $\phi_{k,h} \in \Phi$ is the historical samples till step h of episode k (not including the current step). Let $\Pi := \{\pi : \mathcal{S} \times \Phi \to \mathcal{A}\}$ denote the set of all such policies. Let $I_k(\pi, s, \varphi)$ denote the number of steps taken to reach the terminal state g given the current state s and the historical samples $\varphi \in \Phi$ under the policy $\pi \in \Pi$ in episode k. Let $I_k(\pi^*, s)$ denote the number of steps taken to reach g given the current state s under π^* in episode k. The objective is to find a policy $\pi \in \Pi$ to minimize the expected regret (over K episodes) defined by

$$\mathbb{E}[\mathrm{Reg}_\pi(K)] = \mathbb{E}\left[\sum_{k=1}^{K}\sum_{h=1}^{I_k(\pi^*, S_{k,1})} R_{k,h}(a_1)\right] - \mathbb{E}\left[\sum_{k=1}^{K}\sum_{h=1}^{I_k(\pi, S_{k,1}, \phi_{k,1})} R_{k,h}(\pi(S_{k,h}, \phi_{k,h}))\right].$$

II. Main Results

We propose the ULCB algorithm, which is an index policy like UCB algorithm but the difference is that ULCB uses state-dependent indices. Firstly, the ULCB algorithm plays each arm once by Round-Robin. After that, at step h of episode k, if the state $S_{k,h} = 0$, we let

$$\tilde{\mu}_t^0(a) = \bar{\mu}_t(a) + c_0\sqrt{\frac{\log t + c\log(\log t)}{2N_t(a)}}$$

for all $a \in \mathcal{A}$, where c and c_0 are constants, t is the time step counting from the first episode, $N_t(a) := \sum_{s=1}^{t-1} \mathbb{1}\{A_s = a\}$ denotes the number of times arm a has been pulled before time step t, and $\bar{\mu}_t(a) := \left(\sum_{s=1}^{t-1} \mathbb{1}\{A_s = a\}R_s\right)/N_t(a)$ denotes the average of rewards of pulling arm a before time step t. Note that we also denote the state, the action, and the reward at time step t by S_t, A_t, and R_t, respectively. Then we take an action $A_{k,h} \in \mathrm{argmax}_a\, \tilde{\mu}_t^0(a)$. If the state $S_{k,h} = 1$, we let

$$\tilde{\mu}_t^1(a) = \bar{\mu}_t(a) + c_1\sqrt{\frac{\log t + c\log(\log t)}{2N_t(a)}}$$

for all $a \in \mathcal{A}$, where c_1 is a constant. Then we take an action $A_{k,h} \in \mathrm{argmax}_a\, \tilde{\mu}_t^1(a)$. The algorithm then updates $N_{t+1}(a)$ and $\bar{\mu}_{t+1}(a)$. The process goes to the next step or the next episode depending on whether the abandonment occurs or not.

We also propose the KL-ULCB algorithm, which replaces the indices $\tilde{\mu}_t^0(a)$ and $\tilde{\mu}_t^1(a)$ in ULCB algorithm with

$$\tilde{\mu}_t^0(a) = \min\{p : \mathrm{kl}(\bar{\mu}_t(a), p)N_t(a) \leqslant c_0\log t + c\log(\log t)\}$$
$$\tilde{\mu}_t^1(a) = \max\{p : \mathrm{kl}(\bar{\mu}_t(a), p)N_t(a) \leqslant c_1\log t + c\log(\log t)\}$$

where $\mathrm{kl}(p_1, p_2)$ is the KL divergence between two Bernoulli random variables with parameters p_1 and p_2.

Let $V^*(s)$ and $Q^*(s, a)$ denote the optimal value function and optimal Q-function (their definitions can be found in [1]). We have the following theorem on ULCB.

Theorem 1 (Upper bound for ULCB) *Let Assumption 1 hold. Assume that $q(0,1) < 1$, $q(1,1) < 1$, and for any $a \neq a_1$,*

$$V^*(0) - Q^*(0, a) \geqslant V^*(1) - Q^*(1, a). \tag{1}$$

Then under ULCB with $c_0 = -1$, $c_1 = 1$ and $c = 4$, we have

$$\limsup_{K\to\infty} \frac{\mathbb{E}[\mathrm{Reg}_\pi(K)]}{\log K} \leqslant \sum_{i\neq 1} \frac{V^*(1) - Q^*(1, a_i)}{2(\mu(a_1) - \mu(a_i))^2}.$$

Condition (1) means that a suboptimal pull induces more regret (loss) in state 0 than in state 1. This motivates us to do more exploration in state 1 and to be conservative in state 0, i.e., $c_1 = 1, c_0 = -1$. The instance-dependent constant $\sum_{i\neq 1} \frac{V^*(1)-Q^*(1,a_i)}{2(\mu(a_1)-\mu(a_i))^2}$ in Theorem 1 is smaller than the constant $\sum_{i\neq 1} \frac{V^*(0)-Q^*(0,a_i)}{2(\mu(a_1)-\mu(a_i))^2}$ in the upper bound for the traditional UCB algorithm ($c_0 = c_1 = 1$). For KL-ULCB algorithm, we have the following result.

Theorem 2 (Upper bound for KL-ULCB) *Let all the assumptions in Theorem 1 hold. Then using the KL-ULCB algorithm with $c_0 = c_1 = 1$, and $c = 4$, we have*

$$\limsup_{K\to\infty} \frac{\mathbb{E}[\mathrm{Reg}_\pi(K)]}{\log K} \leqslant \sum_{i\neq 1} \frac{V^*(1) - Q^*(1, a_i)}{\mathrm{kl}(\mu(a_i), \mu(a_1))}.$$

Compared with the result in Theorem 1, the bound in Theorem 2 is better since $\mathrm{kl}(\mu(a_i), \mu(a_1)) \geqslant 2(\mu(a_1) - \mu(a_i))^2$ by Pinsker's inequality. This bound is also better than the one obtained by KL-UCB, $\sum_{i\neq 1} \frac{V^*(0)-Q^*(0,a_i)}{\mathrm{kl}(\mu(a_i),\mu(a_1))}$.

In order to analyze instance-dependent lower bound for MAB-A, similar to [2], [4], we define the set of all consistent policies by Π_{cons}. A policy $\pi \in \Pi_{\mathrm{cons}}$ if for any $\mu(a_1), ..., \mu(a_M)$, $q(0,0)$, $q(0,1)$, $q(1,0)$, $q(1,1)$, and any $\alpha > 0$, $\lim_{K\to\infty} \mathbb{E}[\mathrm{Reg}_\pi(K)]/K^\alpha = 0$. Then Theorem 3 gives an asymptotic lower bound among policies in Π_{cons} for the MAB-A problem.

Theorem 3 (Lower bound) *Let all the assumptions in Theorem 1 hold. For any $\pi \in \Pi_{\mathrm{cons}}$ and any $\mu(a_1)$,...,$\mu(a_M)$, $q(0,0)$, $q(0,1)$, $q(1,0)$, $q(1,1)$ satisfying the assumptions,*

$$\liminf_{K\to\infty} \frac{\mathbb{E}[\mathrm{Reg}_\pi(K)]}{\log K} \geqslant \sum_{i\neq 1} \frac{V^*(1) - Q^*(1, a_i)}{\mathrm{kl}(\mu(a_i), \mu(a_1))}.$$

By Theorem 2 and Theorem 3, the regret upper bound obtained by KL-ULCB attains the lower bound asymptotically.

References

[1] Z. Yang, X. Liu, and L. Ying, "Exploration, exploitation, and engagement in multi-armed bandits with abandonment," *arxiv preprint arxiv.2205.13566*, 2022.

[2] T. Lattimore and C. Szepesvári, *Bandit algorithms*. Cambridge University Press, 2020.

[3] D. P. Bertsekas and J. N. Tsitsiklis, "An analysis of stochastic shortest path problems," *Mathematics of Operations Research*, vol. 16, no. 3, pp. 580–595, 1991.

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