Sequential Multi-armed Bandits

Cem Tekin **Mihaela van der Schaar**
University of California, Los Angeles

Abstract

In this paper we introduce a new class of online learning problems called *sequential multi-armed bandit* (S-MAB) problems. Unlike conventional multi-armed bandit (MAB) problems in which the reward is observed exactly after each taken action, the S-MAB problem proceeds in *rounds*. In each round, the learner sequentially selects from a set of available actions. Upon each action selection a feedback signal is observed, whilst the reward of the selected sequence of actions is only revealed after a *stop* action that ends the current round. The reward of the round depends both on the sequence of actions and the sequence of observed feedbacks. The goal of the learner is to maximize its total expected reward over all rounds by learning to choose the best sequence of actions based on the feedback it gets about these actions. First, we show that combinatorial MABs (C-MAB) are a special case of S-MABs in which the feedback does not matter. Then, we define an *oracle* benchmark, which sequentially selects the actions that maximize the immediate reward. Finally, we propose our online learning algorithm whose regret is logarithmic in time and linear in the number of actions with respect to the oracle benchmark. We evaluate the performance of the proposed model using a personalized online teaching system. Our illustrative results show that online adaptation of the *teaching materials* (actions) based on student feedback can significantly enhance teaching effectiveness.

Preliminary work. Under review by AISTATS 2014. Do not distribute.

1 Introduction

Many sequential decision making problems can be formalized as a MAB problem such as clinical trials [8], dynamic spectrum access [1] and web advertising [9,12]. A common assumption in all these problems is that each decision step involves taking a single action after which the reward is observed. However, unlike these problems in many other applications such as online education [10] and healthcare [11], each decision step involves taking multiple actions for which the reward is only revealed after the action sequence is completed.

For instance, in online education, a sequence of teaching materials are given to the students to improve their understanding of a course subject. While the final exam is used as a benchmark to evaluate the overall effectiveness of the given sequence of teaching materials, a sequence of intermediate feedbacks like students' performance on quizzes, homework grades, etc., can be used to guide the teaching examples online. Similarly, in healthcare a sequence of treatments is given to a patient over a period of time. The overall effectiveness of the treatment plan depends on the given treatments as well as their order [11]. Moreover, the patient can be monitored during the course of the treatment which yields a sequence of feedbacks about the selected treatments, while the final outcome is only available in a follow-up after the treatment is completed.

In conclusion, in such sequential decision making problems the *order* of the taken actions *matters*. Moreover, the feedback available after each taken action drives the action selection process. We call online learning problems exhibiting the aforementioned properties *sequential multi-armed bandits* (S-MAB). An S-MAB problem proceeds in rounds $\rho = 1, 2, \ldots$, in which the learner selects actions sequentially in each round, one after another, with each action belonging to the action set $\mathcal{A}$. After each taken action $a \in \mathcal{A}$, a feedback $f \in \mathcal{F}$ is observed about the taken action. Based on this feedback, the learner either decides to select another action in $\mathcal{A}$ or select a *stop* action which ends the current round and starts the next round. The reward for round ρ is observed only after the stop action is

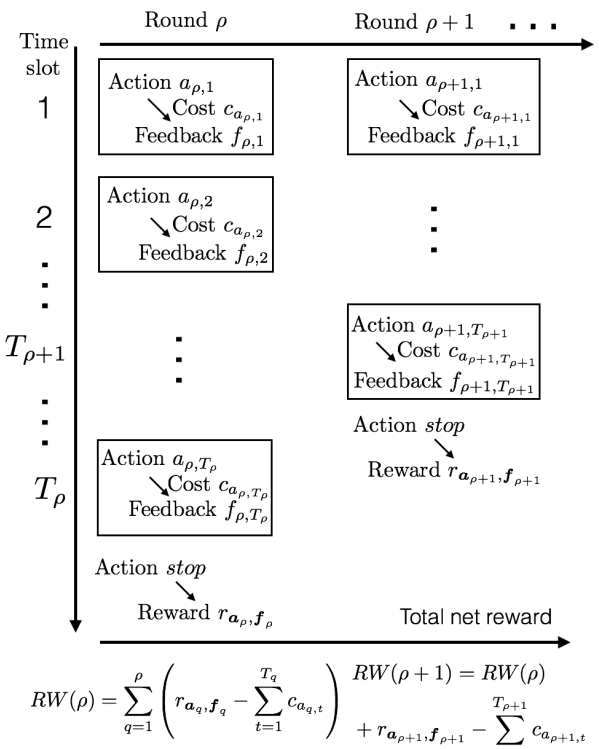

Figure 1: An illustration of the order of actions, feedbacks and rewards for the S-MAB problem. $RW(\rho)$ is the cumulative reward at the end of round ρ, $a_{\rho,t}$ is the tth action selected in round ρ, $f_{\rho,t}$ is the feedback to the tth action in round ρ, and c_a is the cost of selecting action a.

taken. The goal of the learner is to maximize its total expected reward over all rounds by learning to choose the best sequence of actions given the feedback. An illustration that shows the order of actions, feedback, rewards and the rounds is given in Fig. 1.

The S-MAB problem is a generalization of the combinatorial MAB (C-MAB) problem [5]. In a C-MAB problem the learner chooses a sequence of actions and observes both the final reward and the rewards associated with each individual action. The difference from C-MAB problems is that (i) the sequence of actions is not chosen all at once, the chosen sequence depends on the feedback received after each action, (ii) there is no individual reward observation, a (random) *global* reward is observed after the *stop* action is taken, (iii) the reward not only depends on the entire sequence of actions but it also depends on the observed sequence of feedbacks.

For S-MABs we define an oracle benchmark which knows the reward distribution of all sequences of actions and feedbacks, and selects the next action in the sequence to be the action that myopically maximizes the immediate reward. Due to this behavior, we call this benchmark the *best first* (BF) benchmark.[1] We prove the optimality of the BF benchmark for the S-MAB problems with discounted rewards. Then, we introduce a learning algorithm which learns online the actions that myopically maximizes the immediate reward. We prove that the regret of the proposed learning algorithm with respect to the BF benchmark increases logarithmically in the number of rounds.

In standard MAB problems [2,8], the number of possible actions is small, hence one is interested in achieving regret whose time order is small. However, in the S-MAB problem, the number of possible sequences of actions is exponential in the maximum sequence length. Moreover, the set of possible action-feedback sequences is even larger. Due to this, it is very important to design learning algorithms whose regret scales at a much slower rate. Our proposed algorithm's regret is linear in the number of actions times the number of possible feedbacks, which makes it practical in settings where a long sequence of actions should be taken in each round such as healthcare (for each patient) and online education (for each student).

The rest of the paper is organized as follows. Related work is given in Section 2. In Section 3 the S-MAB problem is formalized, the BF benchmark is defined and proven to be optimal for the S-MAB problem with discounted rewards. Then, we propose an online learning algorithm in Section 4 and prove a regret bound for it with respect to the BF benchmark. Illustrative results on the efficiency of the proposed learning algorithm for an online education application is shown in Section 5 based on an online learning platform we developed for students taking a digital signal processing (DSP) class. Finally, concluding remarks are given in Section 6.

2 Related Work

In standard MAB problems [2,8] actions are taken sequentially over time, and a reward is observed after each action is taken, which is then used to update the decision rule. In contrast, in an S-MAB problem, the reward is observed only after a specific *stop* action which ends the current round and starts the next round is taken. This reward depends on the sequence of actions taken and feedbacks observed in the current round. Another class of MAB problems in which the reward depends on the sequence of actions that are taken are the C-MAB problems [5,6]. However, in these problems it is assumed that (i) all the actions in the sequence are selected simultaneously;

[1] Since the BF benchmark chooses the myopic best action it is similar to the best first search algorithms for graphs [14].

hence, no feedback is available between the actions, (*ii*) the *global* reward function has a special *additive* form which is equal to a weighted sum of the *individual* rewards of the selected actions, (*iii*) individual action rewards are observed in addition to the global reward. As we will show in Section 3, the C-MAB problem is a special case of the S-MAB problem.

Other MAB problems which involve large action sets are [4, 7]. In these works, at each time step the learner chooses an action in a metric space and obtains a reward that is a function of the chosen action. Again, no intermediate feedback about the chosen sequence of actions is available before the reward is revealed. Another related MAB problem is MAB with knapsacks [3, 13]. In these problems, there is a limit on the *budget*, which limits the number of times a particular action can be selected. The goal is to maximize the total reward given the budget constraints. However, similar to standard MAB problem, in these problems it is also assumed that the reward is immediately available after each selected action, and the current reward only depends on the current action unlike S-MABs in which the current reward depends on a sequence of actions and feedbacks. Although the S-MAB problem also have a budget constraint which restricts the length of the sequence of actions that can be taken in each round, this constraint is completely different from the budget constraint in MAB with knapsacks. In the S-MAB problem, the budget is renewed after each round; and hence, does not limit the number of rounds in which a certain action can be selected as in MAB with knapsacks.

3 Problem Formulation

The system operates in rounds ($\rho = 1, 2, \ldots$). At each round the learner selects actions sequentially over time until it takes a *stop* action which ends the current round and starts the next round. Let $\mathcal{A}$ denote the set of actions excluding the *stop* action. Let $\overline{\mathcal{A}} := \mathcal{A} \cup \{stop\}$. The number of actions is $A = |\mathcal{A}|$, where $|\cdot|$ is the cardinality operator. After each action is taken a feedback $f \in \mathcal{F}$ is observed about that action. There is a possibility that no feedback is observed, which is denoted by $\emptyset$, hence $\emptyset \in \mathcal{F}$. Let $F = |\mathcal{F}|$. We assume that both sets $\mathcal{A}$ and $\mathcal{F}$ are finite. Upon taking action a the learner incurs a cost $c_a > 0$. Each round ρ is composed of a finite number of slots $t = 1, 2, \ldots, T_\rho$, where T_ρ denotes the number of actions selected before the *stop* action, which is a random variable that depends on the sequence of feedbacks observed in response to the selected actions. We assume that there is a limit on the number of actions that can be selected in each round ρ, i.e., $T_\rho \leq l_{\max}$ for some $l_{\max} > 0$.

For a round ρ, let $a_{\rho,t}$, $1 \leq t \leq T_\rho$ denote the tth action chosen in round ρ, and $f_{\rho,t}$, $1 \leq t \leq T_\rho$ denote the feedback to the tth chosen action in that round. The next action $a_{\rho,t+1} \in \overline{\mathcal{A}}$ may depend on the set of previously selected actions and observed feedbacks. Let $\boldsymbol{a}_\rho := (a_{\rho,1}, \ldots, a_{\rho,T_\rho})$ be the sequence of actions chosen in round ρ. Let $\boldsymbol{a}_\rho[t] := (a_{\rho,1}, \ldots, a_{\rho,t})$ be the sequence of first $t \leq T_\rho$ actions chosen in round ρ. We define $\boldsymbol{f}_\rho$ as the sequence of feedbacks observed about the chosen actions in round ρ and $\boldsymbol{f}_\rho[t]$ as the sequence of first t feedbacks in round ρ.

The set of all sequences of actions is denoted by $\boldsymbol{S}$. Since every sequence of actions must end with the *stop* action we have

$$|\boldsymbol{S}| = \sum_{t=1}^{l_{\max}} A^t = \left(A^{l_{\max}+1} - A\right)/(A-1). \qquad (1)$$

For any sequence of actions $\boldsymbol{a} \in \boldsymbol{S}$, let $\boldsymbol{\mathcal{F}}(\boldsymbol{a})$ be the set of sequences of feedbacks that may be observed, and $\boldsymbol{\mathcal{F}} := \cup_{\boldsymbol{a} \in \boldsymbol{S}} \boldsymbol{\mathcal{F}}(\boldsymbol{a})$. Given a sequence of actions $\boldsymbol{a} \in \boldsymbol{S}$ and sequence of feedbacks $\boldsymbol{f} \in \boldsymbol{\mathcal{F}}(\boldsymbol{a})$ in round ρ, the reward is drawn from an unknown distribution $F_{\boldsymbol{a},\boldsymbol{f}}$ independently from the other rounds. The expected reward is given by $r_{\boldsymbol{a},\boldsymbol{f}}$.

We will first show that the C-MAB problem [5] is a special case of the S-MAB problem.

Definition 1. *In the C-MAB problem, the learner must select M actions (without replacement) for M positions from a set of actions $\mathcal{N}$ such that $|\mathcal{N}| = N \geq M$. The expected reward of action $a \in \mathcal{N}$ in position $t \in \{1, \ldots, M\}$ is given by $\theta_{a,t} > 0$. For a sequence of actions $\boldsymbol{a} = (a_1, \ldots, a_M)$, where a_t denotes the action assigned to the tth position, the expected reward is given by $\theta_{\boldsymbol{a}} = \sum_{t=1}^{M} \theta_{a_t,t}$.*

The next theorem shows that the C-MAB problem is a special case of the S-MAB problem.

Theorem 1. *Consider the C-MAB problem given in Definition 1. Define an S-MAB problem with $\mathcal{A} = \mathcal{N}$, $\mathcal{F} = \emptyset$, $c_a = 0$, for all $a \in \mathcal{A}$, $l_{\max} = M$ and*

$$r_{\boldsymbol{a},\boldsymbol{f}} = \sum_{t=1}^{|\boldsymbol{a}|} \theta_{a_t,t} \mathrm{I}(a_t \neq a_{t'} \text{ for } t \neq t'),$$

for $\boldsymbol{a}$ such that $|\boldsymbol{a}| = M$ and for all $\boldsymbol{f} \in \boldsymbol{\mathcal{F}}(\boldsymbol{a})$, where $\mathrm{I}(\cdot)$ is the indicator function; and $r_{\boldsymbol{a},\boldsymbol{f}} = 0$ for all $\boldsymbol{a}$ such that $|\boldsymbol{a}| < M$. These two problems are equivalent in the sense that every sequence of feasible actions $\boldsymbol{a}$ in the S-MAB problem, i.e., sequence of actions with nonzero reward, has the same reward in the C-MAB problem.

Proof. $\mathcal{F} = \emptyset$ implies that the feedback has no effect on the reward, hence the reward only depends on the sequence of taken actions. For all $\boldsymbol{a}$ such that $|\boldsymbol{a}| <$

M or $\boldsymbol{a}$ such that the same action is taken in two or more positions, the reward in the S-MAB problem is zero. These sequences of actions are not feasible since $\theta_{a,t} > 0$ for all $a \in \mathcal{N}$ and $t = 1, \ldots, M$ guarantees that at least one sequence of actions have a positive reward. For any sequence of actions $\boldsymbol{a}$ and sequence of feedbacks $\boldsymbol{f}$ such that $|\boldsymbol{a}| = M$, we have $r_{\boldsymbol{a},\boldsymbol{f}} = \sum_{t=1}^{M} \theta_{a_t,t} = \theta_{\boldsymbol{a}} > 0$. □

3.1 The Best First Benchmark

Since the number of possible sequences of actions and feedbacks is exponential in $l_{\max}$, it is very inefficient to learn the best sequence of actions by trying each of them separately to estimate $r_{\boldsymbol{a},\boldsymbol{f}}$ for every $\boldsymbol{a} \in \boldsymbol{S}$ and $\boldsymbol{f} \in \boldsymbol{\mathcal{F}}(\boldsymbol{a})$. In this section we propose an oracle benchmark called the best first (BF) benchmark whose action selection strategy can be learned quickly by the learner. The pseudocode for the BF benchmark is given in Fig. 2.

```
1: while ρ ≥ 1 do
2:    Select action a*_1 = arg max_{a∈A} y_{a,∅}
3:    Observe feedback f*_1.
4:    while 1 < t ≤ l_max do
5:       if
         r_{a*[t−1],f*[t−1]} ≥
         max_{a∈A}(y_{(a*[t−1],a),f*[t−1]} − c_a) then
6:          a*_t = stop  //BREAK
         else
8:          a*_t = arg max_{a∈A}(y_{(a*[t−1],a),f*[t−1]} − c_a)
         end if
         t = t + 1
11:   end while
12:   ρ = ρ + 1
13: end while
```

Figure 2: Pseudocode for the BF benchmark.

Let $\boldsymbol{S}[t] \subset \boldsymbol{S}$ be the set of sequences of actions of length t followed by the *stop* action. When we need to explicitly state the length of the chosen sequence of actions, we will use the notation $\boldsymbol{a}[t] \in \boldsymbol{S}[t]$. We will also use $\boldsymbol{f}_{\boldsymbol{a}}[t']$ to denote the sequence of feedbacks to the first t' actions in $\boldsymbol{a}$. Let

$$y_{\boldsymbol{a}[t],\boldsymbol{f}_{\boldsymbol{a}[t]}[t-1]} := \mathrm{E}_f[r_{\boldsymbol{a}[t],(\boldsymbol{f}_{\boldsymbol{a}[t]}[t-1],f)}],$$

be the *ex-ante* reward given the sequence of actions $\boldsymbol{a}[t]$ before the feedback for a_t is observed, where the expectation is taken with respect to the distribution of the feedback for action a_t.

The BF benchmark incrementally selects the next action based on the sequence of feedbacks observed for the previously selected actions. The action it selects is $a_1^* = \arg\max_{a \in \mathcal{A}} y_{a,\emptyset}$, where $\emptyset$ denotes that no previous feedback is available. Let $\boldsymbol{a}^* = (a_1^*, a_2^*, \ldots, a_T^*)$ be the sequence of actions selected by the BF benchmark, where T is the random time slot in which the *stop* action is selected, which obviously depends on the observed sequence of feedbacks. In general a_t^*, depends on both $\boldsymbol{a}^*[t-1]$ and $f_{\boldsymbol{a}^*[t-1]}[t-1]$. We assume that the following property holds for the expected reward for the sequence of actions selected by the BF benchmark and the sequence of feedbacks observed from these actions.

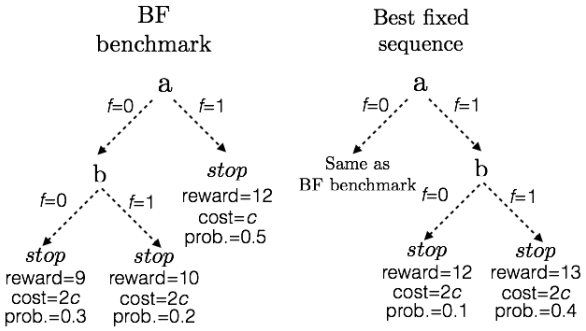

Figure 3: Decision graph for the BF benchmark and the best fixed sequence of actions.

Assumption 1. *For any two sequences of action-feedback pairs $(\boldsymbol{a}^*, \boldsymbol{f})$ and $(\boldsymbol{a}^*, \boldsymbol{f}')$, where $\boldsymbol{a}^*$ is the set of actions selected by the BF benchmark and $\boldsymbol{f}$ and $\boldsymbol{f}'$ are two feedback sequences that are associated with this set of actions, if $f_t = f_t'$, then we have*

$$\arg\max_{a \in \mathcal{A}} y_{(\boldsymbol{a}^*[t],a),\boldsymbol{f}[t]} = \arg\max_{a \in \mathcal{A}} y_{(\boldsymbol{a}^*[t],a),\boldsymbol{f}'[t]}.$$

For any t, if $r_{\boldsymbol{a}^*[t],\boldsymbol{f}^*[t]} \geq y_{(\boldsymbol{a}^*[t],a),\boldsymbol{f}^*[t]} - c_a$ for all $a \in \mathcal{A}$, then the BF benchmark will select the *stop* action after the tth action. Otherwise, it will select the action which maximizes $y_{(\boldsymbol{a}^*[t],a),\boldsymbol{f}^*[t]} - c_a$. The total expected *net reward*, i.e., the expected total reward minus costs of choosing actions, of the BF benchmark for the first n rounds is equal to

$$RW_{\mathrm{BF}}(n) := \sum_{\rho=1}^{n} \mathrm{E}\left[Y_{\boldsymbol{A}_\rho^*,\boldsymbol{F}_\rho^*} - \sum_{a \in \boldsymbol{A}_\rho^*} c_a \right],$$

where $\boldsymbol{A}_\rho^*$ is the random variable that represents the sequence of actions selected in round ρ by the BF benchmark, $\boldsymbol{F}_\rho^*$ is the random variable that represents the sequence of feedbacks observed for the actions selected in round ρ, and $Y_{\boldsymbol{A}_\rho^*\boldsymbol{F}_\rho^*}$ is the random variable that represents the reward obtained in round ρ.

Although the BF benchmark may not always select the optimal sequence actions, it can perform better than the best fixed sequence of actions that is not adapted based on the observed feedbacks. This is illustrated in the following example.

Example 1. *Consider $\mathcal{A} = \{a, b\}$, $\mathcal{F} = \{0, 1\}$, $c_a = c_b = c$ and $l_{\max} = 2$. Assume that the expected rewards are given as follows: $r_{(a,a)|(f_1,f_2)} = 0$ and $r_{(b,b)|(f_1,f_2)} = 0$ for any $f_1, f_2 \in \mathcal{F}$; $r_{a,0} = 0$, $r_{b,0} = 0$, $r_{a,1} = 12$, $r_{b,1} = 6$, $r_{(a,b),(1,1)} = 13$, $r_{(a,b),(1,0)} = 12$, $r_{(a,b),(0,1)} = 10$, $r_{(a,b),(0,0)} = 9$. Let*

$\mathrm{P}(\boldsymbol{f}|\boldsymbol{a})$ *denote the probability that feedback sequence* $\boldsymbol{f}$ *is observed for the sequence of actions* $\boldsymbol{a}$*. Assume that we have* $\mathrm{P}(1|a) = 0.5$*,* $\mathrm{P}(0|a) = 0.5$*,* $\mathrm{P}((0,0)|(a,b)) = 0.3$*,* $\mathrm{P}((0,1)|(a,b)) = 0.2$*,* $\mathrm{P}((1,1)|(a,b)) = 0.4$*,* $\mathrm{P}((1,0)|(a,b)) = 0.1$*.*

The decision rules of the BF benchmark and the best fixed sequence of actions are shown in Figure 3. The BF benchmark selects a *as the first action. Then, if feedback is* 0 *it selects* b *before selecting the stop action. Else, it selects the stop action after* a*. Hence the expected reward of the BF benchmark in a single round is*

$$RW_{BF}(1) = 0.5 \times 12 + 0.3 \times (9 - c) + 0.2 \times (10 - c) - c = 10.7 - 1.5c.$$

The best fixed sequence of actions is (a, b) *which gives a single round expected reward that is equal to*

$$0.3 \times 9 + 0.4 \times 12 + 0.2 \times 10 + 0.1 \times 11 - 2c = 11 - 2c.$$

Thus, for $c > 3/5$ *the BF benchmark is better than the best fixed sequence of actions.*

Although the BF benchmark is not optimal in general, there exists a special class of S-MAB problems for which it is optimal. We call these problems, S-MAB with *discounted rewards*.

Definition 2. *S-MAB with discounted rewards is an S-MAB problem whose expected reward for any sequence of actions* $\boldsymbol{a}$ *and feedbacks* $\boldsymbol{f}$ *is given by* $r_{\boldsymbol{a},\boldsymbol{f}} = \sum_{t=1}^{|\boldsymbol{a}|} \delta^{t-1}\theta_{a_t,f_t}$*, where* $0 \leq \delta < 1$ *is the discount factor, and* $\theta_{a,f} \geq 0$ *is the expected reward of action* a *when feedback to action* a *is* f*.*

In the next theorem we show that the BF benchmark is optimal for the S-MAB problem with discounted rewards.

Theorem 2. *The BF benchmark is optimal for the S-MAB problem with discounted rewards for any discount factor* $0 \leq \delta < 1$*.*

Proof. Consider any sequence of actions $\boldsymbol{a}[t]$ and feedbacks $\boldsymbol{f}[t]$. Given these, the optimal action to select at $t+1$ is $\arg\max_{a\in\mathcal{A}}(\mathrm{E}_f[\theta_{a,f}] - c_a)$ if $\delta^t \mathrm{E}_f[\theta_{a,f}] > c_a$ for some $a \in \mathcal{A}$. Otherwise the optimal action is the *stop* action. This implies that for all $1 \leq t \leq l_{\max}$, the optimal action coincides with the action chosen by the BF benchmark. □

3.2 Definition of the Regret

Consider any learning algorithm α which selects a sequence of actions $\boldsymbol{A}_\rho^\alpha$ based on the observed sequence of feedbacks $\boldsymbol{F}_\rho^\alpha$. The regret of α with respect to the BF benchmark in the first n rounds is given by

$$\mathrm{E}[R(n)] := RW_{BF}(n) - \sum_{\rho=1}^{n} \mathrm{E}\left[Y_{\boldsymbol{A}_\rho^\alpha \boldsymbol{F}_\rho^\alpha} - \sum_{a\in\boldsymbol{A}_\rho^\alpha} c_a \right]. \quad (2)$$

Any algorithm whose regret increases at most sublinearly, i.e., $O(n^\gamma)$, $0 < \gamma < 1$, in the number of rounds will converge in terms of the average reward to the average reward of the BF benchmark as $n \to \infty$. In the next section we will propose an algorithm whose regret increases only logarithmically (better than sublinear) in the number of rounds.

4 A Learning Algorithm for the S-MAB Problem

In this section we propose *Feedback Adaptive Learning* (FAL) (pseudocode given in Fig. 4), which learns the sequence of actions to select based on the observed feedbacks to the previous actions (as shown in Fig. 1). In order to minimize the regret given in (2), FAL balances exploration and exploitation when selecting the actions. Consider the tth action selected in round ρ. FAL keeps the following sample mean reward estimates: (i) $\hat{r}_{t,a,f}(\rho)$ which is the sample mean estimate of the rewards collected in the first $\rho - 1$ rounds in which the *stop* action is taken after action a is selected as the tth action and feedback f is observed, (ii) $\hat{y}_{f,t,a}(\rho)$ which is the sample mean estimate of the rewards in the first $\rho - 1$ rounds in which the *stop* action is taken after action a is selected as the tth action after observing feedback f for the $t-1$th action. In addition to these, FAL keeps the following counters: (i) $T_{t,a,f}(\rho)$ which counts the number of times action a is selected as the tth action and feedback f is observed in the first $\rho - 1$ rounds in which the *stop* action is taken after the tth action, (ii) $T_{f,t,a}(\rho)$ which counts the number of times action a is selected as the tth action after feedback f is observed from the previously selected action in the first $\rho - 1$ rounds in which the *stop* action is taken after the tth action.

Next, we explain how exploration and exploitation is performed. Consider the event that FAL selects action $a_{\rho,t} = a$ and receives feedback $f_{\rho,t} = f$. It first checks if $T_{t,a,f}(\rho) < D\log(\rho/\delta)$, where $D > 0$ and $\delta > 0$ are constants that are input parameters of FAL whose values will be specified later. If this holds, then FAL explores by taking the *stop* action and obtaining the reward $Y(\rho)$, by which it updates $\hat{r}_{t,a,f}(\rho+1) = (\hat{r}_{t,a,f}(\rho) + Y(\rho))/(T_{t,a,f}(\rho) + 1)$. Else if $T_{t,a,f}(\rho) \geq D\log(\rho/\delta)$, FAL checks if there are any actions $a' \in \mathcal{A}$ for which $T_{f_{\rho,t},t+1,a'}(\rho) < D\log(\rho/\delta)$. If there are such actions, then FAL randomly selects one of them to explore by observing the feedback, and then taking the *stop* action, and obtaining the reward. The obtained rewad $Y(\rho)$ is used for both updating $\hat{r}_{t+1,a',f_{\rho,t+1}}(\rho+1)$ and $\hat{y}_{f_{\rho,t},t+1,a'}(\rho+1)$. If none of the above events happen, then FAL exploits at step t. To do this it first checks if $\hat{r}_{t,a,f_{\rho,t}}(\rho) \geq \hat{y}_{f_{\rho,t},t+1,a'}(\rho) - c_{a'}$, for all $a' \in \mathcal{A}$. If this is the case, it means that selecting one more action does not increase the expected

Input $D > 0$, $\delta > 0$, $\mathcal{A}$, $\mathcal{F}$, $l_{\max}$.
Initialize: $\hat{r}_{t,a,f} = 0, \hat{y}_{f,t,a} = 0$, $T_{t,a,f} = 0$, $T_{f,t,a} = 0$, $\forall a \in \mathcal{A}, f \in \mathcal{F}, t = 1, \ldots, l_{\max}$. $f_{\rho,0} = \emptyset$, $\boldsymbol{a}_\rho[0] = \emptyset$, $\forall \rho = 1, 2, \ldots$.
while $\rho \geq 1$ **do**
  $\mathcal{U}_1 = \{a \in \mathcal{A} : T_{\emptyset,1,a} < D \log(\rho/\delta)\}$
  if $\mathcal{U}_1 \neq \emptyset$ **then**
    Select $a_{\rho,1}$ randomly from $\mathcal{U}_1$, observe $f_{\rho,1}$.
    Select the *stop* action, get reward $Y(\rho)$, $t^* = 1$, //BREAK
  else
    Select $a_{\rho,1} = \arg\max_{a \in \mathcal{A}}(\hat{y}_{\emptyset,1,a} - c_a)$, observe $f_{\rho,1}$.
  end if
  $t = 2$
  while $2 \leq t \leq l_{\max}$ **do**
    $\mathcal{U}_t = \{a \in \mathcal{A} : T_{f_{\rho,t-1},t,a} < D \log(\rho/\delta)\}$
    if $T_{t-1,a_{\rho,t-1},f_{\rho,t-1}} < D \log(\rho/\delta)$ **then**
      Select the *stop* action, get reward $Y(\rho)$, $t^* = t - 1$, //BREAK
    else if $\mathcal{U}_t \neq \emptyset$ **then**
      Select $a_{\rho,t}$ randomly from $\mathcal{U}_t$ and observe the feedback $f_{\rho,t}$.
      Select the *stop* action, get reward $Y(\rho)$, $t^* = t$, //BREAK
    else
      if $\hat{r}_{t-1,a_{\rho,t-1},f_{\rho,t-1}} \geq \hat{y}_{f_{\rho,t-1},t,a'} - c_a$, $\forall a' \in \mathcal{A}$ **then**
        Select the *stop* action, get reward $Y(\rho)$, $t^* = t - 1$, //BREAK
      else
        Select $a_{\rho,t} = \arg\max_{a' \in \mathcal{A}}(\hat{y}_{f_{\rho,t-1},t,a'} - c_{a'}$ and get the feedback $f_{\rho,t}$.
      end if
    end if
    $t = t + 1$
  end while
  Update $\hat{r}_{t^*,a_{\rho,t^*},f_{\rho,t^*}}$, $\hat{y}_{f_{\rho,t^*-1},t^*,a_{\rho,t*}}$ using $Y(\rho)$ (sample mean update).
  $T_{t^*,a_{\rho,t^*},f_{\rho,t^*}} ++$, $T_{f_{\rho,t^*-1},t^*,a_{\rho,t^*}} ++$.
  $\rho = \rho + 1$
end while

Figure 4: Pseudocode for FAL.

reward enough to compensate for the cost associated with selecting one more action. Hence, FAL takes the *stop* action after the tth action. If the opposite case happens, then it means that selecting one more action can improve the reward sufficiently enough for it to compensate the cost of selecting the action. Hence, FAL will select one more action which is given by $a_{\rho,t+1} = \arg\max_{a' \in \mathcal{A}}(\hat{y}_{f_{\rho,t},t+1,a'}(\rho) - c_{a'})$. The next decision to take (whether to select another action in $\mathcal{A}$ or to select the *stop* acion) will be based on the feedback to $a_{\rho,t+1}$ which is $f_{\rho,t+1}$. This goes on until FAL takes the *stop* action, which will eventually happen since at most $l_{\max}$ actions can be taken in a round. This way the length of the sequence of selected actions is adapted based on the sequence of received feedbacks and costs of taking the actions. Since FAL's objective is to maximize the net reward (reward minus costs of selecting actions) from a sequence of actions, it captures the tradeoff between the rewards and the costs in selecting its actions.

4.1 Regret Bound For FAL

The regret of FAL can be written as the sum of two separate regret terms: total regret in rounds when FAL explores, i.e., $R_e(n)$, and total regret in rounds when FAL exploits, i.e., $R_s(n)$. Hence, we can write $\mathrm{E}[R(n)] = \mathrm{E}[R_e(n)] + \mathrm{E}[R_s(n)]$. In order to derive the regret bound for FAL, we assume that the following holds.

Assumption 2. ***Unique optimal action for every history of sequence of actions and feedbacks.*** *Let* $Q_1^* := \arg\max_{a \in \mathcal{A}} y_{a,\emptyset}$, *and for any* $\boldsymbol{a}[t] \in \boldsymbol{S}[t]$ *and* $\boldsymbol{f}[t] \in \boldsymbol{\mathcal{F}}(\boldsymbol{a}[t])$, $t \geq 1$ *let*

$$Q_{t+1}^*(\boldsymbol{a}[t], \boldsymbol{f}[t]) := \arg\max_{a \in \bar{\mathcal{A}}} \left\{ r_{(\boldsymbol{a}[t],stop),\boldsymbol{f}[t]}, \left\{ y_{(\boldsymbol{a}[t],a'),\boldsymbol{f}[t]} - c_{a'} \right\}_{a' \in \mathcal{A}} \right\},$$

where $r_{(\boldsymbol{a}[t],stop),\boldsymbol{f}[t]} := r_{\boldsymbol{a}[t],\boldsymbol{f}[t]}$. *We assume that* $|Q_1^*| = 1$ *and* $|Q_{t+1}^*(\boldsymbol{a}[t], \boldsymbol{f}[t])| = 1$ *for all* $\boldsymbol{a}[t] \in \boldsymbol{S}[t]$ *and* $\boldsymbol{f}[t] \in \boldsymbol{\mathcal{F}}(\boldsymbol{a}[t])$, $1 \leq t \leq l_{\max} - 1$.

For a sequence of numbers $\{r\}_{r \in \mathcal{R}}$, let $\mathrm{min2}(\{r\}_{r \in \mathcal{R}})$ be the difference between the highest and the second highest numbers. Consider any sequence of actions $\boldsymbol{a}^*[t] \in \boldsymbol{S}[t]$ and feedback $\boldsymbol{f}[t] \in \boldsymbol{\mathcal{F}}(\boldsymbol{a}^*[t])$, where $\boldsymbol{a}^*[t]$ is the sequence of the first t actions selected by the BF benchmark. Let $\Delta_{\min,1} := \mathrm{min2}(\{y_{a,\emptyset}\}_{a \in \mathcal{A}})$, and

$$\Delta_{\min,t} := \min_{\boldsymbol{a}^*[t] \in \boldsymbol{S}[t], \boldsymbol{f}[t] \in \boldsymbol{\mathcal{F}}(\boldsymbol{a}^*[t])} \left(\mathrm{min2}(r_{\boldsymbol{a}^*[t],\boldsymbol{f}[t]}, \{y_{(\boldsymbol{a}^*[t],a),\boldsymbol{f}[t]} - c_a\}_{a \in \mathcal{A}}) \right),$$

for $1 < t < l_{\max}$. Let $\Delta_{\min} := \min_{t=1,\ldots,l_{\max}-1} \Delta_{\min,t}$. Given that the constant D that is input to FAL is such that $D \geq 4/\Delta_{\min}^2$, and assuming that the support set of the rewards is $[0, 1]$, we have the following bounds on the regret.

Theorem 3. *Setting the parameters of FAL as* $D \geq 4/\Delta_{\min}^2$ *and* $\delta = \sqrt{\epsilon}/(A\sqrt{2\beta})$, *where* $\beta = \sum_{t=1}^{\infty} 1/t^2$, *we have the following bounds on the regret of FAL.*
(i) $R_e(n) \leq 2 l_{\max} FAD \log(n/\delta)$ *with probability 1.*
(ii) $R_s(n) = 0$ *with probability at least* $1 - \epsilon$.
(iii) $\mathrm{E}[R(n)] \leq 2 l_{\max} FAD \log(n/\delta) + \epsilon n$.

Proof. The proof involves showing that when the FAL estimates the expected rewards for the sequences of actions it selects such that they are within $\Delta_{\min}/2$ of the true expected rewards, then it will always select the same sequence of actions as the BF benchmark does.

To proceed, we define the following sets of rounds. Let $E_1(n)$ be the set of rounds in $\{1, \ldots, n\}$ for which FAL explores the action selected in the first slot of that round, i.e., $\rho \in \{1, \ldots, n\}$ for which $T_{\emptyset,1,a}(\rho) < D \log(\rho/\delta)$ for some $a \in \mathcal{A}$ such that after this action, the *stop* action is taken. Let $E_t(n)$, $1 < t \leq l_{\max}$

be the set of rounds in $\{1,\dots,n\}$ for which FAL explores in the tth slot of that round, i.e., the set of rounds for which FAL exploited up to the $t-1$th slot and $\hat{r}_{t-1,a_{\rho,t-1},f_{\rho,t-1}}(\rho) \leq D\log(\rho/\delta)$ or $\hat{y}_{f_{\rho,t-1},t,a} \leq D\log(\rho/\delta)$ for some $a \in \mathcal{A}$ such that the *stop* action is taken either after the $t-1$th slot or the tth slot depending on which action in which slot is under-explored. Let $\tau_1(n)$ be the set of rounds in $\{1,\dots,n\}$ for which FAL exploits for the first slot of the round, i.e., $T_{\emptyset,1,a} \geq D\log(\rho/\delta)$ for all $a \in \mathcal{A}$. Let $\tau_t(n)$ be the set of rounds in $\{1,\dots,n\}$ for which FAL exploits for the tth slot in that round, i.e., $\hat{r}_{t-1,a_{\rho,t-1},f_{\rho,t-1}}(\rho) \geq D\log(\rho/\delta)$ and $\hat{y}_{f_{\rho,t-1},t,a} \geq D\log(\rho/\delta)$ for all $a \in \mathcal{A}$ such that FAL has not taken the *stop* action before slot $t-1$. Let $Z_t(n) := \tau_t(n) - \tau_{t+1}(n) - E_{t+1}(n)$, $1 \leq t < l_{\max}$ denote the set of rounds in $\{1,\dots,n\}$ for which FAL takes the *stop* action after the $t-1$th action is selected at rounds when it exploits. Let $Z_{l_{\max}}(n) := \tau_{l_{\max}}(n)$. The set of all rounds for which FAL explores until the nth round is equal to $E(n) := \bigcup_{t=1}^{l_{\max}} E_1(n)$, where $E_t(n) \cap E_{t'}(n) = \emptyset$ for $t \neq t'$. The set of all rounds for which FAL exploits until the nth round is equal to $Z(n) := \bigcup_{t=1}^{l_{\max}} Z_t(n)$, where $Z_t(n) \cap Z_{t'}(n) = \emptyset$ for $t \neq t'$. We also have $Z(n) := \tau_1(n)$, $\tau_t(n) = \tau_{t+1}(n) \cup E_{t+1}(n) \cup Z_t(n)$ for $1 \leq t < l_{\max}$.

In the following we will bound $R_s(n)$. We define the events which correspond to the case that the estimated rewards for the sequences of actions that will also be selected by the BF benchmark are within $\Delta_{\min}/2$ of the expected final exam scores. Let $\boldsymbol{a}^*_\rho$ be the sequence of actions that will be selected by the BF benchmark in round ρ. Let

$$\begin{aligned}&\mathrm{Perf}_1(n)\\ &:= \{|\hat{y}_{\emptyset,1,a}(\rho) - y_{a,\emptyset}| < \Delta_{\min}/2, \forall a \in \mathcal{A}, \forall \rho \in \tau_1(n)\},\end{aligned}$$

and

$$\begin{aligned}\mathrm{Perf}_t(n) := \Big\{&|\hat{r}_{t-1,a_{\rho,t-1},f_{\rho,t-1}}(\rho) - r_{\boldsymbol{a}^*_\rho[t-1],\boldsymbol{f}_{\boldsymbol{a}^*_\rho[t-1]}[t-1]}|\\ &< \Delta_{\min}/2, |\hat{y}_{f_{\rho,t-1},t,a}(\rho) - y_{(\boldsymbol{a}^*_\rho[t-1],a),\boldsymbol{f}_{\boldsymbol{a}^*_\rho[t-1]}[t-1]}|\\ &< \Delta_{\min}/2, \forall a \in \mathcal{A}, \forall \rho \in \tau_t(n)\}\end{aligned}$$

Let $\mathrm{Perf}(n) = \bigcap_{t=1}^{l_{\max}} \mathrm{Perf}_t(n)$. On event $\mathrm{Perf}(n)$, FAL selects sequence of actions in the same way as the BF benchmark does. Hence, the contribution to the regret given in (2) on event $\mathrm{Perf}(n)$ is zero.

Next, we lower bound the probability of event $\mathrm{Perf}(n)$. Using the chain rule we can write

$$\begin{aligned}&\mathrm{P}(\mathrm{Perf}(n)) = \mathrm{P}(\mathrm{Perf}_{l_{\max}}(n), \mathrm{Perf}_{l_{\max}-1}(n), \dots, \mathrm{Perf}_1(n))\\ &= \mathrm{P}(\mathrm{Perf}_{l_{\max}}(n)|\mathrm{Perf}_{l_{\max}-1}(n), \dots, \mathrm{Perf}_1(n))\\ &\times \mathrm{P}(\mathrm{Perf}_{l_{\max}-1}(n)|\mathrm{Perf}_{l_{\max}-2}(n), \dots, \mathrm{Perf}_1(n))\\ &\times \dots \times \mathrm{P}(\mathrm{Perf}_2(n)|\mathrm{Perf}_1(n)) \times \mathrm{P}(\mathrm{Perf}_1(n)). \qquad (3)\end{aligned}$$

For an event E, let E^c denote its complement. Note that we have

$$\begin{aligned}\mathrm{P}(\mathrm{Perf}_1(n)^c) &\leq \sum_{\rho\in\tau_1(n)} \sum_{a\in\mathcal{A}} \mathrm{P}(|\hat{y}_{\emptyset,1,a}(\rho) - y_{a,\emptyset}| < \Delta_{\min}/2)\\ &\leq \sum_{\rho\in\tau_1(n)} 2A\exp(-2D\log(\rho/\delta)\Delta^2_{\min}/4)\\ &\leq \sum_{\rho\in\tau_1(n)} 2A\delta^2/\rho^2 \leq 2A\beta\delta^2, \qquad (4)\end{aligned}$$

since $D \geq 4/\Delta^2_{\min}$ and $\beta = \sum_{\rho=1}^{\infty} 1/\rho^2$. Hence, we have $\mathrm{P}(\mathrm{Perf}_1(n)) \geq 1 - 2A\beta\delta^2$. On event $\mathrm{Perf}_1(n)$, it is always the case that the first selected action by FAL is chosen according to the BF benchmark, independent of whether the FAL explores or exploits in the second slot of those rounds. Hence given $\mathrm{Perf}_1(n)$, the sample mean reward estimates that are related to $\mathrm{Perf}_2(n)$ are always sampled from the distribution in which the first action is selected according to the BF benchmark. Because of this, we have $\mathrm{P}(\mathrm{Perf}_2(n)|\mathrm{Perf}_1(n)) \geq 1 - 2A\beta\delta^2$. Similarly, it can be shown that $\mathrm{P}(\mathrm{Perf}_t(n)|\mathrm{Perf}_{t-1}(n), \dots, \mathrm{Perf}_1(n)) \geq 1 - 2A\beta\delta^2$. Combining all of this and using (4) we get

$$\mathrm{P}(\mathrm{Perf}(n)) \geq (1 - 2A\beta\delta^2)^A \geq 1 - 2A^2\beta\delta^2 = 1 - \epsilon, \qquad (5)$$

since $\delta = \sqrt{\epsilon}/(A\sqrt{2\beta})$.

Next we bound $R_e(n)$. From the definition of $E_t(n)$, $t = 1,\dots,l_{\max}$, we know that $|E_1(n)| \leq FAD\log(n/\delta)$. Similarly, for $E_t(n)$, $t = 2,\dots,l_{\max}$, we have $|E_t(n)| \leq 2FAD\log(n/\delta)$. Hence, we have $|E(n)| \leq 2l_{\max}FAD\log(n/\delta)$. Since the worst-case reward loss due to a suboptimal sequence of actions is at most 1, we have $R_e(n) \leq 2l_{\max}FAD\log(n/\delta)$. Finally, the regret bound on $\mathrm{E}[R(n)]$ holds by taking the expectation. □

Theorem 3 provides bounds on the exploration, exploitation and the total regret of FAL. The regret bounds are in the order of $l_{\max}A$, which is significantly lower than $A^{l_{\max}}$ which is the order of the number of sequences of actions as given in (1). This implies that FAL learns to exploit the actions that are selected by the BF benchmark much faster than standard MAB algorithms [2, 8], whose rates of exploration for the problem we consider will be in the order of $A^{l_{\max}}$. The learner can set ϵ to a desired value based on the number of times it wants to explore and the confidence level it wants to achieve for the rounds that it exploits. The following corollary gives a logarithmic in the number of rounds bound on the regret, which is achieved for a specific horizon n by setting the value of $\epsilon = 1/n$.

Corollary 1. *Given the number of rounds n as an input, setting the parameters of FAL as $D = 4/\Delta^2_{\min}$*

and $\delta = 1/(A\sqrt{2n\beta})$*, where* $\beta = \sum_{t=1}^{\infty} 1/t^2$*, we have*

$$\mathrm{E}[R(n)] \leq 1 + 2l_{\max}FAD\log(A\sqrt{2\beta}) + l_{\max}FAD\log(n).$$

Proof. The result is obtained by setting $\epsilon = 1/n$ and using the results of Theorem 3. □

Corollary 1 implies that the expected total loss of FAL with respect to the BF benchmark grows at most logarithmically in the number of rounds. Hence as $n \to \infty$, the average regret converges to zero.

5 Illustrative Results

In this section we illustrate the performance of FAL in a real-world online education platform that we designed for remedial teaching. We call this implementation of FAL – eTutor, and its operation is shown in Fig. 5. We deployed our eTutor system for remedial studies for students who have already studied digital signal processing (DSP) one or more years ago, and the goal of this implementation of the eTutor is to refresh *discrete Fourier transform* (DFT) material in the minimum amount of time. In order to learn what sequence of teaching materials (actions) is best to show to the students, we differentiate students according to their contexts. Student contexts belong to $\mathcal{X} = \{0, 1\}$, where for a student ρ, $x_\rho = 0$ implies that she is not confident about her knowledge of DFT, and $x_\rho = 1$ implies that she is confident about her knowledge of DFT. For each context we run a different instance of the FAL algorithm to learn the best sequence of teaching materials to show to the students based on the feedback they give. $\mathcal{A}$ contains three (*remedial*) materials: one text that describes DFT and two questions that refreshes DFT knowledge. If a question is shown to the student and if the student's answer is incorrect, then the correct answer is shown along with an explanation. For each $a \in \mathcal{A}$, we set the cost to be $c_a = \lambda \times \theta_a$, where θ_a (in minutes) is the average time it takes for a student to complete material a, and $0 < \lambda < 1$ is a constant that represents the tradeoff between time and final exam score. For instance, in remedial teaching, reducing the time it takes to teach a concept is as important as improving the final exam score. The value of θ_a is estimated and updated based on the responses of the students. Performance (reward) of the students after taking the remedial materials are tested by the same final exam.

We compare the performance of eTutor with a *random rule* (RR) that randomly selects the materials to show and a *fixed rule* (FR) that shows all materials (text first, easy question second, hard question third).[2] The

[2]Since the other MAB algorithms are not adaptive to the feedbacks, they cannot do better than the best fixed rule in the long run.

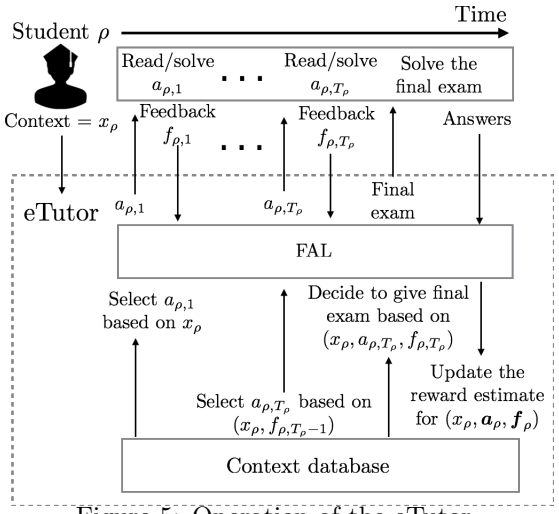

Figure 5: Operation of the eTutor.

Algorithm	average final score (max=100)	time spent in minutes taking the course
eTutor	75.8	8.5
RR	62.5	10.2
FR	75.0	17.0

Table 1: Comparison of eTutor with RR and FR.

average final score achieved by these algorithms for $n = 500$ students and $\lambda = 0.04$ is shown in Table 1. From this table we see that eTutor achieves 15.7% and 1.1% improvement in the average final score for $n = 500$ compared to RR and FR, respectively. The improvement compared to FR is small because FR shows all the materials to every student. The average time spent by each student taking the course is 8.5 minutes for eTutor which is 16.7% and 50% less than the average time it takes for the same set of students by RR and FR, respectively. eTutor achieves significant savings in time by showing the best materials to each student based on her context instead of showing everything to every student.

6 Conclusion

In this paper, we proposed a new class of online learning problems called sequential multi-armed bandits. Although the number of possible sequences of actions increases exponentially with the length of the sequence, we proved that efficient online learning algorithms which have regret that grows linearly with the number of actions and logarithmically in time exist. We illustrate the proposed S-MAB model in an online education platform. Possible future research directions include applying S-MAB problems to other settings such as healthcare. For instance, an effective sequence of actions that maximizes the patients' recovery rate can be learned in medical treatments using FAL.

References

[1] A. Anandkumar, N. Michael, and A.K. Tang. Opportunistic spectrum access with multiple players: Learning under competition. In *Proc. of IEEE INFOCOM*, March 2010.

[2] P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47:235–256, 2002.

[3] A. Badanidiyuru, R. Kleinberg, and A. Slivkins. Bandits with knapsacks. In *Foundations of Computer Science (FOCS), 2013 IEEE 54th Annual Symposium on*, pages 207–216, 2013.

[4] N. Cesa-Bianchi and G. Lugosi. Combinatorial bandits. *Journal of Computer and System Sciences*, 78(5):1404–1422, 2012.

[5] Y. Gai, B. Krishnamachari, and R. Jain. Learning multiuser channel allocations in cognitive radio networks: A combinatorial multi-armed bandit formulation. In *Proc. of the IEEE Symposium on New Frontiers in Dynamic Spectrum*, pages 1–9, 2010.

[6] Y. Gai, B. Krishnamachari, and R. Jain. Combinatorial network optimization with unknown variables: Multi-armed bandits with linear rewards and individual observations. *IEEE/ACM Transactions on Networking (TON)*, 20(5):1466–1478, 2012.

[7] R. Kleinberg, A. Slivkins, and E. Upfal. Multi-armed bandits in metric spaces. In *Proc. of the 40th annual ACM symposium on Theory of Computing*, pages 681–690, 2008.

[8] T. Lai and H. Robbins. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6:4–22, 1985.

[9] L. Li, W. Chu, J. Langford, and R.E. Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proc. of the 19th international conference on World wide web*, pages 661–670. ACM, 2010.

[10] S. Piramuthu. Knowledge-based web-enabled agents and intelligent tutoring systems. *Education, IEEE Transactions on*, 48(4):750–756, Nov 2005.

[11] A.J. Schaefer, M.D. Bailey, S.M. Shechter, and M.S. Roberts. Modeling medical treatment using Markov decision processes. In *Operations Research and Health Care*, pages 593–612. Springer, 2004.

[12] A. Slivkins. Contextual bandits with similarity information. In *Proc. of the 24th Annual Conference on Learning Theory (COLT)*, 2011.

[13] L. Tran-Thanh, A.C. Chapman, A. Rogers, and N.R. Jennings. Knapsack based optimal policies for budget-limited multi-armed bandits. In *AAAI*, 2012.

[14] N.R. Vempaty, V. Kumar, and R.E. Korf. Depth-first versus best-first search. In *AAAI*, pages 434–440, 1991.