Regret Bounds for Sleeping Experts and Bandits

Robert Kleinberg[*]
Department of Computer Science
Cornell University
Ithaca, NY 14853
rdk@cs.cornell.edu

Alexandru Niculescu-Mizil[†]
Department of Computer Science
Cornell University
Ithaca, NY 14853
alexn@cs.cornell.edu

Yogeshwer Sharma[‡]
Department of Computer Science
Cornell University
Ithaca, NY 14853
yogi@cs.cornell.edu

Abstract

We study on-line decision problems where the set of actions that are available to the decision algorithm vary over time. With a few notable exceptions, such problems remained largely unaddressed in the literature, despite their applicability to a large number of practical problems. Departing from previous work on this "Sleeping Experts" problem, we compare algorithms against the payoff obtained by the *best ordering* of the actions, which is a natural benchmark for this type of problem. We study both the full-information (best expert) and partial-information (multi-armed bandit) settings and consider both stochastic and adaptive adversaries. For all settings we give algorithms achieving (almost) information-theoretically optimal regret bounds (up to a constant or a sub-logarithmic factor) with respect to the best-ordering benchmark.

1 Introduction

In on-line decision problems, or sequential prediction problems, an algorithm must choose, in each of the T consecutive rounds, one of the n possible actions. In each round, each action receives a real valued positive payoff in $[0, 1]$, initially unknown to the algorithm. At the end of each round the algorithm is revealed some information about the payoffs of the actions in that round. The goal of the algorithm is to maximize the total payoff, i.e. the sum of the payoffs of the chosen actions in each round. The standard on-line decision settings are the *best expert* setting (or the full-information setting) in which, at the end of the round, the payoffs of *all* n strategies are revealed to the algorithm, and the *multi-armed bandit* setting (or the partial-information setting) in which only the payoff of the chosen strategy is revealed. Customarily, in the best expert setting the strategies are called *experts* and in the multi-armed bandit setting the strategies are called *bandits* or *arms*. We use *actions* to generically refer to both types of strategies, when we do not refer particularly to either.

The performance of the algorithm is typically measured in terms of *regret*. The regret is the difference between the expected payoff of the algorithm and the payoff of a single fixed strategy for selecting actions. The usual single fixed strategy to compare against is the one which always selects the expert or bandit that has the highest total payoff over the T rounds (in hindsight).

The usual assumption in online learning problems is that all actions are available at all times. In many applications, however, this assumption is not appropriate. In network routing problems, for example, some of the routes are unavailable at some point in time due to router or link crashes. Or, in electronic commerce problems, items are out of stock, sellers are not available (due to maintenance or simply going out of business), and buyers do not buy all the time. Even in the setting that originally motivated the multi-armed bandit problems, a gambler playing slot machines, some of the slot machines might be occupied by other players at any given time.

In this paper we relax the assumption that all actions are available at all times, and allow the set of available actions to vary from one round to the next, a model known as "predictors that specialize" or "sleeping experts" in prior work. The first foundational question that needs to be addressed is how to define regret when the set of available actions may vary over time. Defining regret with respect to the best action in hindsight is no longer appropriate since that action might sometimes be unavailable. A useful thought experiment for guiding our intuition is the following: if each action had a fixed payoff distribution that was *known* to the decision-maker, what would be the best way to choose among the available actions? The answer is obvious: one should order all of the actions according to their expected payoff, then choose among the available actions by selecting the one which ranks highest in this ordering. Guided by the outcome of this thought experiment, we define our base to be the best ordering of actions in hindsight (see Section 2 for a formal definition) and contend that this is a natural and intuitive way to define regret in our setting. This contention is also supported by the informal observation that order-based decision rules seem to resemble the way people make choices in situations with a varying set of actions, e.g. choosing which brand of beer to buy at a store.

We prove lower and upper bounds on the regret with re-

[*] Supported by NSF grants CCF-0643934 and CCF-0729102.

[†] Supported by NSF grants 0347318, 0412930, 0427914, and 0612031.

[‡] Supported by NSF grant CCF-0514628.

spect to the best ordering for both the best expert setting and the multi-armed bandit settings. We first explore the case of stochastic adversary, where the payoffs received by expert (bandit) i at each time step are independent samples from an unknown but fixed distribution $P_i(\cdot)$ supported on $[0, 1]$ with mean μ_i. Assuming that $\mu_1 > \mu_2 > \cdots > \mu_n$ (and the algorithm, of course, does not know the identities of these actions) we show that the regret of any learning algorithm will necessarily be at least $\Omega\left(\sum_{i=1}^{n-1} \frac{1}{\mu_i - \mu_{i+1}}\right)$ in the best expert setting, and $\Omega\left(\log(T) \sum_{i=1}^{n-1} \frac{1}{\mu_i - \mu_{i+1}}\right)$ in the multi-armed bandit setting if the game is played for T rounds (for T sufficiently large). We also present efficient learning algorithms for both settings. For the multi-armed bandit setting, our algorithm, called AUER, is an adaptation of the UCB1 algorithm in Auer et al [ACBF02], which comes within a constant factor of the lower bound mentioned above. For the expert setting, a very simple algorithm, called "follow-the-awake-leader", which is a variant of "follow-the-leader" [Han57, KV05], comes within a constant factor of the lower bound above. While our algorithms are adaptations of existing techniques, the proofs of the upper and lower bounds hinge on some technical innovations. For the lower bound, we must modify the classic asymptotic lower bound proof of Lai and Robbins [LR85] to obtain a bound which holds at all sufficiently large finite times. We also prove a novel lemma (Lemma 3) that allows us to relate a regret upper bound arising from application of UCB1 to a sum of lower bounds for two-armed bandit problems.

Next we explore the fully adversarial case where we make no assumptions on how the payoffs for each action are generated. We show that the regret of any learning algorithm must be at least $\Omega\left(\sqrt{Tn\log(n)}\right)$ for the best expert setting and $\Omega\left(\sqrt{Tn^2}\right)$ for the multi-armed bandit setting. We also present algorithms whose regret is within a constant factor of the lower bound for the best expert setting, and within $\mathcal{O}\left(\sqrt{\log(n)}\right)$ of the lower bound for the multi-armed bandit setting. It is worth noting that the gap of $\mathcal{O}\left(\sqrt{\log n}\right)$ also exists in the all-awake bandit problem.

The fully adversarial case, however, proves to be harder, and neither algorithm is computational efficient. To appreciate the hardness of the fully adversarial case, one can prove[1] that, unless $\mathrm{P} = \mathrm{NP}$, any low regret algorithm that learns internally a consistent ordering over experts can not be computationally efficient. Note that this does not mean that there can be no computationally efficient, low regret algorithms for the fully adversarial case. There might exist learning algorithms that are able to achieve low regret without actually learning a consistent ordering over experts. Finding such algorithms, if they do indeed exist, remains an open problem.

1.1 Related work

Sequential prediction problems. The best-expert and multi-armed bandit problems correspond to special cases of our model in which every action is always available. These problems have been widely studied, and we draw on this literature to design algorithms and prove lower bounds for the generalizations considered here. The adversarial expert paradigm was introduced by Littlestone and Warmuth [LW94], and Vovk [Vov90]. Cesa-Bianchi et al [CBFH$^+$97] further developed this paradigm in work which gave optimal regret bounds of $\sqrt{T(\ln n)}$ and Vovk [Vov98] characterized the achievable regret bounds in these settings.

The multi-armed bandit model was introduced by Robbins [Rob]. Lai and Robbins [LR85] gave asymptotically optimal strategies for the stochastic version of bandit problem—in which there is a distribution of rewards on each arm and the rewards in each time step are drawn according to this distribution. Auer, Cesa-Bianchi, Fischer [ACBF02] introduced the algorithm UCB1 and showed that the optimal regret bounds of $\mathcal{O}(\log T)$ can be achieved uniformly over time for the stochastic bandit problem. (In this bound, the big-O hides a constant depending on the means and differences of means of payoffs.) For the adversarial version of the multi-armed bandit problem, Auer, Cesa-Bianchi, Freund, and Schapire [ACBFS02] proposed the algorithm Exp3 which achieves the regret bound of $\mathcal{O}(\sqrt{Tn\log n})$, leaving a $\sqrt{\log n}$ factor gap from the lower bound of $\Omega(\sqrt{nT})$. It is worth noting that the lower bound holds even for an oblivious adversary, one which chooses a sequence of payoff functions independently of the algorithm's choices.

Prediction with sleeping experts. Freund, Schapire, Singer, and Warmuth [FSSW97] and Blum and Mansour [BM05] have considered sleeping experts problems before, analyzing algorithms in a framework different from the one we adopt here. In the model of Freund et al., as in our model, a set of awake experts is specified in each time period. The goal of the algorithm is to choose one expert in each time period so as to minimize regret against the best "mixture" of experts (which constitutes their benchmark). A mixture $\mathbf{u}$ is a probability distribution $(u_1, u_2, \ldots, u_n)$ over n experts which in time period t selects an expert according to the restriction of $\mathbf{u}$ to the set of awake experts.

We consider a natural evaluation criterion, namely the best ordering of experts. In the special case when all experts are always awake, both evaluation criteria degenerate to picking the best expert. Our "best ordering" criterion can be regarded as a degenerate case of the "best mixture" criterion of Freund et al. as follows. For the ordering σ, we assign probabilities $\frac{1}{Z}(1, \epsilon, \epsilon^2, \ldots, \epsilon^{n-1})$ to the sequence of experts $(\sigma(1), \sigma(2), \ldots, \sigma(n))$ where $Z = \frac{1-\epsilon^n}{1-\epsilon}$ is the normalization factor and $\epsilon > 0$ is an arbitrarily small positive constant. The only problem is that the bounds that we get from [FSSW97] in this degenerate case are very weak. As $\epsilon \to 0$, their bound reduces to comparing the algorithm's performance to the ordering σ's performance only for time periods when $\sigma(1)$ expert is awake, and ignoring the time periods when $\sigma(1)$ is not awake. Therefore, a natural reduction of our problem to the problem considered by Freund et al. defeats the purpose of giving equal importance to all time periods.

Blum and Mansour [BM05] consider a generalization of the sleeping expert problem, where one has a set of *time selection functions* and the algorithm aims to have low regret

[1] It is a simple reduction from feedback arc set problem, which is omitted from this extended abstract.

with respect to every expert, according to every time selection function. It is possible to solve our regret-minimization problem (with respect to the best ordering of experts) by reducing to the regret-minimization problem solved by Blum and Mansour, but this leads to an algorithm which is neither computationally efficient nor information-theoretically optimal. We now sketch the details of this reduction. One can define a time selection function for each (ordering, expert) pair (σ, i), according to $I_{\sigma,i}(t) = 1$ if $i \preceq_\sigma j$ for all $j \in A_t$ (that is, σ chooses i in time period t if $I_{\sigma,i}(t) = 1$). The regret can now be bounded, using Blum and Mansour's analysis, as

$$\sum_{i=1}^{n} \mathcal{O}\left(\sqrt{T_i \log(n \cdot n! \cdot n)} + \log(n! \cdot n^2)\right) = \mathcal{O}\left(\sqrt{Tn^2 \log n} + n \log n\right).$$

This algorithm takes exponential time (due to the exponential number of time selection functions) and gives a regret bound of $\mathcal{O}(\sqrt{Tn^2 \log n})$ against the best ordering, a bound which we improve in Section 4 using a different algorithm which also takes exponential time but is information-theoretically optimal. (Of course, Blum and Mansour were designing their algorithm for a different objective, not trying to get low regret with respect to best ordering. Our improved bound for regret with respect to the best ordering does not imply an improved bound for experts learning with time selection functions.)

A recent paper by Langford and Zhang [LZ07] presents an algorithm called the *Epoch-Greedy algorithm* for bandit problems with side information. This is a generalization of the multi-armed bandit problem in which the algorithm is supplied with a piece of *side information* in each time period before deciding which action to play. Given a hypothesis class $\mathcal{H}$ of functions mapping side information to actions, the Epoch-Greedy algorithm achieves low regret against a sequence of actions generated by applying a single function $h \in \mathcal{H}$ to map the side information in every time period to an action. (The function h is chosen so that the resulting sequence has the largest possible total payoff.) The stochastic case of our problem is reducible to theirs, by treating the set of available actions, A_t, as a piece of side information and considering the hypothesis class $\mathcal{H}$ consisting of functions h_σ, for each total ordering σ of the set of actions, such that $h_\sigma(A)$ selects the element of A which appears first in the ordering σ. The regret bound in [LZ07] is expressed implicitly in terms of the expected regret of an empirical reward maximization estimator, which makes it difficult to compare this bound with ours. Instead of pursuing this reduction from our problem to the contextual bandit problem in [LZ07], Section 3.1.1 presents a very simple bandit algorithm for the stochastic setting with an explicit regret bound that is provably information-theoretically optimal.

2 Terminology and Conventions

We assume that there is a fixed pool of actions, $\{1, 2, ...n\}$, with n known. We will sometimes refer to an action by *expert* in the best expert setting and by *arm* or *bandit* in the multi-armed bandit setting. At each time step $t \in \{1, 2, ..., T\}$, an adversary chooses a subset $A_t \subseteq \{1, 2, ..., n\}$ of the actions to be available. The algorithm can only choose among available actions, and only available actions receive rewards. The reward received by an available action i at time t is $r_i(t) \in [0, 1]$.

We will consider two models for assigning rewards to actions: a stochastic model and an adversarial model. (In contrast, the choice of the set of awake experts is always adversarial.) In the stochastic model the reward for arm i at time t, $r_i(t)$, is drawn independently from a fixed unknown distribution $P_i(\cdot)$ with mean μ_i. In the adversarial model we make no stochastic assumptions on how the rewards are assigned to actions. Instead, we assume that the rewards are selected by an adversary. The adversary is potentially but not necessarily randomized.

Let σ be an ordering (permutation) of the n actions, and A a subset of the actions. We denote by $\sigma(A)$ the action in A that is highest ranked in σ. The reward of an ordering is the reward obtained by selecting at each time step the highest ranked action available.

$$R_{\sigma,T} = \sum_{t=1}^{T} r_{\sigma(A_t)}(t) \tag{1}$$

Let $R_T = \max_\sigma R_{\sigma,T}$ ($\max_\sigma \mathbb{E}[R_{\sigma,T}]$ in the stochastic rewards model) be the reward obtained by the best ordering. We define the regret of an algorithm with respect to the best ordering as the expected difference between the reward obtained by the best ordering and the total reward of the algorithm's chosen actions $x(1), x(2), ..., x(t)$:

$$REG_T = \mathbb{E}\left[R_T - \sum_{t=1}^{T} r_{x(t)}(t)\right] \tag{2}$$

where the expectation is taken over the algorithm's random choices and the randomness of the reward assignment in the stochastic reward model.

3 Stochastic Model of Rewards

We first explore the stochastic rewards model, where the reward for action i at each time step is drawn independently from a fixed unknown distribution $P_i(\cdot)$ with mean μ_i. For simplicity of presentation, throughout this section we assume that $\mu_1 > \mu_2 > \cdots > \mu_n$. That is the lower numbered actions are better than the higher numbered actions. Let $\Delta_{i,j} = \mu_i - \mu_j$ for all $i < j$ be the expected increase in the reward of expert i over expert j.

We present optimal (up to a constant factor) algorithms for both the best expert and the multi-armed bandit setting. Both algorithms are natural extensions of algorithms for the all-awake problem to the sleeping-experts problem. The analysis of the algorithms, however, is not a straightforward extension of the analysis for the all-awake problem and new proof techniques are required.

3.1 Best expert setting

In this section we study algorithms for the best expert setting with stochastic rewards. We prove matching (up to a constant factor) information-theoretic upper and lower bounds on the regret of such algorithms.

3.1.1 Upper bound (algorithm: FTAL)

To get an upper bound on regret we adapt the "follow the leader" algorithm [Han57, KV05] to the sleeping experts setting: at each time step the algorithm chooses the awake expert that has the highest average payoff, where the average is taken over the time steps when the expert was awake. If an expert is awake for the first time, then the algorithm chooses it. (If there are more than one such such experts, then the algorithm chooses one of them arbitrarily.) The pseudocode for the algorithm is shown in Algorithm 1. The algorithm is called **F**ollow **T**he **A**wake **L**eader (FTAL for short).

```
Initialize $z_i = 0$ and $n_i = 0$ for all $i \in [n]$.
for $t = 1$ to $T$ do
    if $\exists j \in A_t$ s.t. $n_j = 0$ then
        Play expert $x(t) = j$
    else
        Play expert $x(t) = \arg\max_{i \in A_t} \left(\frac{z_i}{n_i}\right)$
    end
    Observe payoff $r_i(t)$ for all $i \in A_t$
    $z_i \leftarrow z_i + r_i(t)$ for all $i \in A_t$
    $n_i \leftarrow n_i + 1$ for all $i \in A_t$
end
```

Algorithm 1: Follow-the-awake-leader (FTAL) algorithm for sleeping experts problem with stochastic adversary.

Theorem 1 *The* FTAL *algorithm has a regret of at most*

$$\sum_{j=1}^{n-1} \frac{32}{\Delta_{j,j+1}}$$

with respect to the best ordering.

The theorem follows immediately from the following pair of lemmas. The second of these lemmas will also be used in Section 3.2.

Lemma 2 *The* FTAL *algorithm has a regret of at most*

$$\sum_{j=2}^{n} \sum_{i=1}^{j-1} \frac{8}{\Delta_{i,j}^2} (\Delta_{i,i+1} + \Delta_{j-1,j})$$

with respect to the best ordering.

Proof: Let $n_{i,t}$ be the number of times expert i has been awake until time t. Let $\hat{\mu}_{i,t}$ be expert i's average payoff until time t. The Azuma-Hoeffding Inequality [Azu67, Hoe63] says that

$$\begin{aligned}\mathbb{P}[n_{j,t}\hat{\mu}_{j,t} > n_{j,t}\mu_j + n_{j,t}\Delta_{i,j}/2] \\ \leq e^{-\frac{n_{j,t}^2 \Delta_{i,j}^2}{8 \cdot n_{j,t}}} = e^{-\frac{\Delta_{i,j}^2 n_{j,t}}{8}},\end{aligned}$$

and

$$\begin{aligned}\mathbb{P}[n_{i,t}\hat{\mu}_{i,t} < n_{i,t}\mu_i - n_{i,t}\Delta_{i,j}/2] \\ \leq e^{-\frac{n_{i,t}^2 \Delta_{i,j}^2}{8 \cdot n_{i,t}}} = e^{-\frac{\Delta_{i,j}^2 n_{i,t}}{8}}.\end{aligned}$$

Let us say that the FTAL algorithm suffers an (i,j)-*anomaly of type 1* at time t if $x_t = j$ and $\hat{\mu}_{j,t} - \mu_j > \Delta_{i,j}/2$. Let us say that FTAL suffers an (i,j)-*anomaly of type 2* at time t if $i_t^* = i$ and $\mu_i - \hat{\mu}_{i,t} > \Delta_{i,j}/2$. Note that when FTAL picks a strategy $x_t = j \neq i = i_t^*$, it suffers an (i,j)-anomaly of type 1 or 2, or possibly both. We will denote the event of an (i,j)-anomaly of type 1 (resp. type 2) at time t by $\mathcal{E}_{i,j}^{(1)}(t)$ (resp. $\mathcal{E}_{i,j}^{(2)}(t)$, and we will use $M_{i,j}^{(1)}$, resp. $M_{i,j}^{(2)}$, to denote the total number of (i,j)-anomalies of types 1 and 2, respectively. We can bound the expected value of $M_{i,j}^{(1)}$ by

$$\begin{aligned}\mathbb{E}[M_{i,j}^{(1)}] &\leq \sum_{t=1}^{\infty} e^{-\frac{\Delta_{i,j}^2 n_{j,t}}{8}} \mathbf{1}\{j \in A_t\} \qquad (3) \\ &\leq \sum_{n=1}^{\infty} e^{-\frac{\Delta_{i,j}^2 n}{8}} \qquad (4) \\ &= \frac{1}{e^{\Delta_{i,j}^2/8} - 1} \leq \frac{8}{\Delta_{i,j}^2},\end{aligned}$$

where line (4) is justified by observing that distinct nonzero terms in (3) have distinct values of $n_{j,t}$. The expectation of $M_{i,j}^{(2)}$ is also bounded by $8/\Delta_{i,j}^2$, via an analogous argument.

Recall that A_t denotes the set of awake experts at time t, $x_t \in A_t$ denotes the algorithm's choice at time t, and $r_i(t)$ denotes the payoff of expert i at time t (which is distributed according to $P_i(\cdot)$). Let $i_t^* \in A_t$ denote the optimal expert at time t (i.e., the lowest-numbered element of A_t). Let us bound the regret of the FTAL algorithm now.

$$\begin{aligned}&\mathbb{E}\left[\sum_{t=1}^{T} \left(r_{i_t^*}(t) - r_{x_t}(t)\right)\right] \\ &= \mathbb{E}\left[\sum_{t=1}^{T} \Delta_{i_t^*, x_t}\right] \\ &= \mathbb{E}\left[\sum_{t=1}^{T} \mathbf{1}\left\{\mathcal{E}_{i_t^*, x_t}^{(1)}(t) \vee \mathcal{E}_{i_t^*, x_t}^{(2)}(t)\right\} \Delta_{i_t^*, x_t}\right] \\ &\leq \mathbb{E}\left[\sum_{t=1}^{T} \mathbf{1}\left\{\mathcal{E}_{i_t^*, x_t}^{(1)}(t)\right\} \Delta_{i_t^*, x_t}\right] \\ &\quad + \mathbb{E}\left[\sum_{t=1}^{T} \mathbf{1}\left\{\mathcal{E}_{i_t^*, x_t}^{(2)}(t)\right\} \Delta_{i_t^*, x_t}\right]\end{aligned}$$

With the convention that $\Delta_{i,j} = 0$ for $j \leq i$, the first term can be bounded by:

$$\begin{aligned}&\mathbb{E}\left[\sum_{t=1}^{T} \mathbf{1}\left\{\mathcal{E}_{i_t^*, x_t}^{(1)}(t)\right\} \Delta_{i_t^*, x_t}\right] \\ &= \mathbb{E}\left[\sum_{t=1}^{T} \sum_{j=2}^{n} \mathbf{1}\left\{\mathcal{E}_{i_t^*, j}^{(1)}(t)\right\} \Delta_{i_t^*, j}\right]\end{aligned}$$

(Since the event $\mathcal{E}_{i_t^*, j}^{(1)}(t)$ occurs only for $j = x_t$.)

$$= \mathbb{E}\left[\sum_{t=1}^{T} \sum_{j=2}^{n} \mathbf{1}\left\{\mathcal{E}_{i_t^*, j}^{(1)}(t)\right\} \sum_{i=i_t^*}^{j-1} (\Delta_{i,j} - \Delta_{i+1,j})\right] \qquad (5)$$

$$\leq \mathbb{E}\left[\sum_{t=1}^{T}\sum_{j=2}^{n}\sum_{i=i_t^*}^{j-1} \mathbf{1}\left\{\mathcal{E}_{i,j}^{(1)}(t)\right\}\Delta_{i,i+1}\right]$$

(Since $\mathbf{1}\left\{\mathcal{E}^{(1)}_{i_1,j}(t)\right\} \leq \mathbf{1}\left\{\mathcal{E}^{(1)}_{i_2,j}(t)\right\}$ for all $i_1 \leq i_2 < j$.)

$$\begin{aligned}
&\leq \mathbb{E}\left[\sum_{j=2}^{n}\sum_{i=1}^{j-1}\Delta_{i,i+1}\sum_{t=1}^{T}\mathbf{1}\left\{\mathcal{E}_{i,j}^{(1)}(t)\right\}\right]\\
&= \sum_{j=2}^{n}\sum_{i=1}^{j-1}\Delta_{i,i+1}\mathbb{E}[M_{i,j}^{(1)}]\\
&\leq \sum_{1\leq i<j\leq n}\frac{8}{\Delta_{i,j}^2}\Delta_{i,i+1}.
\end{aligned}$$

Similarly, the second term can be bounded by

$$\begin{aligned}
&\mathbb{E}\left[\sum_{t=1}^{T}\mathbf{1}\left\{\mathcal{E}_{i_t^*,x_t}^{(2)}(t)\right\}\Delta_{i_t^*,x_t}\right]\\
&= \mathbb{E}\left[\sum_{t=1}^{T}\sum_{i=1}^{n-1}\mathbf{1}\left\{\mathcal{E}_{i,x_t}^{(2)}(t)\right\}\Delta_{i,x_t}\right]
\end{aligned}$$

(Since event $\mathcal{E}^{(2)}_{i,x_t}(t)$ occurs only for $i = i_t^*$.)

$$= \mathbb{E}\left[\sum_{t=1}^{T}\sum_{i=1}^{n-1}\mathbf{1}\left\{\mathcal{E}_{i,x_t}^{(2)}(t)\right\}\sum_{j=i+1}^{x_t}(\Delta_{i,j}-\Delta_{i,j-1})\right] \tag{6}$$

$$\leq \mathbb{E}\left[\sum_{t=1}^{T}\sum_{i=1}^{n-1}\sum_{j=i+1}^{x_t}\mathbf{1}\left\{\mathcal{E}_{i,j}^{(2)}(t)\right\}\Delta_{j-1,j}\right]$$

(Since $\mathbf{1}\left\{\mathcal{E}^{(2)}_{i,j_1}(t)\right\} \geq \mathbf{1}\left\{\mathcal{E}^{(2)}_{i,j_2}(t)\right\}$ for all $i < j_1 \leq j_2$.)

$$\begin{aligned}
&\leq \mathbb{E}\left[\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\Delta_{j-1,j}\sum_{t=1}^{T}\mathbf{1}\left\{\mathcal{E}_{i,j}^{(2)}(t)\right\}\right]\\
&= \sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\Delta_{j-1,j}\mathbb{E}[M_{i,j}^{(2)}]\\
&\leq \sum_{1\leq i<j\leq n}\frac{8}{\Delta_{i,j}^2}\Delta_{j-1,j}
\end{aligned}$$

Adding the two bounds gives the statement of the lemma. ■

Lemma 3 *For $\Delta_{i,j} = \mu_i - \mu_j$ defined as above*

$$\sum_{1\leq i<j\leq n}\Delta_{i,j}^{-2}\Delta_{i,i+1} \leq 2\sum_{j=2}^{n}\Delta_{j-1,j}^{-1}$$

and

$$\sum_{1\leq i<j\leq n}\Delta_{i,j}^{-2}\Delta_{j-1,j} \leq 2\sum_{j=2}^{n}\Delta_{j-1,j}^{-1}.$$

Proof: It suffices to prove the first of the two inequalities stated in the lemma; the second follows from the first by replacing each μ_i with $1-\mu_i$, which has the effect of replacing $\Delta_{i,j}$ with $\Delta_{n+1-j,n+1-i}$.

For a fixed $i \in [n]$, we write $\sum_{j:j>i}\Delta_{i,j}^{-2}$ as follows.

$$\begin{aligned}
\sum_{j:j>i}\Delta_{i,j}^{-2} &= \sum_{j=2}^{n}\mathbf{1}\{j>i\}\Delta_{i,j}^{-2} \qquad (7)\\
&= \int_{x=0}^{\infty}\#\left\{j : j>i, \Delta_{i,j}^{-2}\geq x\right\}\mathrm{d}x\\
&= \int_{x=0}^{\infty}\#\left\{j>i, \Delta_{i,j}\leq x^{-1/2}\right\}\mathrm{d}x\\
&= -2\int_{y=\infty}^{0}\#\{j>i, \Delta_{i,j}\leq y\}\,y^{-3}\mathrm{d}y
\end{aligned}$$

(Changing the variable of integration $x^{-1/2} = y$)

$$= 2\int_{y=0}^{\infty}\#\{j>i, \Delta_{i,j}\leq y\}\,y^{-3}\mathrm{d}y. \tag{8}$$

Let us make the following definition, which will be used in the proof below.

Definition 4 *For an expert j and $y \geq 0$, let $i_y(j)$ be the minimum numbered expert $i \leq j$ such that $\Delta_{i,j}$ is no more than y. That is*

$$i_y(j) := \arg\min\{i : i\leq j, \Delta_{i,j}\leq y\}.$$

Now we can write the following chain of inequalities. (Note that the best (highest payoff) expert is indexed as 1, and lowest payoff is indexed n.)

$$\begin{aligned}
&\sum_{j=2}^{n}\sum_{i=1}^{j-1}\Delta_{i,j}^{-2}\Delta_{i,i+1} \qquad (9)\\
&= \sum_{i=1}^{n-1}\Delta_{i,i+1}\sum_{j:j>i}\Delta_{i,j}^{-2}\\
&= 2\sum_{i=1}^{n-1}\Delta_{i,i+1}\left(\int_{y=0}^{\infty}\#\{j : j>i, \Delta_{i,j}\leq y\}\,y^{-3}\mathrm{d}y\right)
\end{aligned}$$

(From (8).)

$$= 2\int_{y=0}^{\infty}y^{-3}\left(\sum_{i=1}^{n-1}\Delta_{i,i+1}\cdot\#\{j>i, \Delta_{i,j}\leq y\}\right)\mathrm{d}y$$

(Changing the order of integration and summation.)

$$= 2\int_{y=0}^{\infty}y^{-3}\left(\sum_{i=1}^{n-1}\Delta_{i,i+1}\sum_{j=i+1}^{n}\mathbf{1}\{j>i, \Delta_{i,j}\leq y\}\right)\mathrm{d}y$$

(Expanding $\#\{\cdot\}$ into sum of $\mathbf{1}\{\cdot\}$.)

$$= 2\int_{y=0}^{\infty}y^{-3}\left(\sum_{j=2}^{n}\sum_{i=1}^{j-1}\Delta_{i,i+1}\mathbf{1}\{j>i, \Delta_{i,j}\leq y\}\right)\mathrm{d}y$$

(Changing the order of summation.) Recall from Definition 4 that for any j and $y \geq 0$, $i_y(j)$ is the least indexed expert i such that $\Delta_{i,j}$ is still less than y. We get the following.

$$= 2\int_{y=0}^{\infty} y^{-3}\left(\sum_{j=2}^{n}\sum_{i=i_y(j)}^{j-1}\Delta_{i,i+1}\right)\mathrm{d}y$$

$$= 2\int_{y=0}^{\infty} y^{-3}\left(\sum_{j=2}^{n}\left(\mu_{i_y(j)}-\mu_j\right)\right)\mathrm{d}y$$

$$= 2\sum_{j=2}^{n}\int_{y=0}^{\infty} y^{-3}\left(\mu_{i_y(j)}-\mu_j\right)\mathrm{d}y$$

(Changing the order of summation and integration.)

$$= 2\sum_{j=2}^{n}\int_{y=\Delta_{j-1,j}}^{\infty} y^{-3}\left(\mu_{i_y(j)}-\mu_j\right)\mathrm{d}y \qquad (10)$$

(This is because for values of y less than $\Delta_{j-1,j}$, $i_y(j) = j$ and integrand is equal to zero.)

$$\leq 2\sum_{j=2}^{n}\int_{y=\Delta_{j-1,j}}^{\infty} y^{-3}\cdot y\,\mathrm{d}y$$

(Since $\mu_{i_y(j)} - \mu_j \leq y$.)

$$= 2\sum_{j=2}^{n}\int_{y=\Delta_{j-1,j}}^{\infty} y^{-2}\mathrm{d}y$$

$$= 2\sum_{j=2}^{n}\Delta_{j-1,j}^{-1} \qquad (11)$$

This concludes the proof of the lemma. ■

Remarks for small $\Delta_{i,i+1}$ Note that the upper bound stated in Theorem 1 become very large when $\Delta_{i,i+1}$ is very small for some i. Indeed, when mean payoffs of all experts are equal, $\Delta_{i,i+1} = 0$ for all i and upper bound becomes trivial, while the algorithm does well (picking any expert is as good as any other). We suggest a slight modification of the proof to take care of such case.

Let $\epsilon > 0$ be fixed (the original theorem corresponds to the case $\epsilon = 0$). Recall the definition of $i_\epsilon(j)$ from Definition 4. We also define the inverse, $j_\epsilon(i)$ as the maximum numbered expert j such that $\Delta_{i,j}$ is no more than ϵ, i.e., $j_\epsilon(i) = \arg\max\{j : j \geq i, \Delta_{i,j} \leq \epsilon\}$. Note that the three conditions: (1) $i < i_\epsilon(j)$, (2) $j > j_\epsilon(i)$, and (3) $\Delta_{i,j} > \epsilon$ are equivalent. The idea in this new analysis is to "identify" experts that have means within ϵ of each other. (We cannot just make equivalence classes based on this, since the relation of "being within ϵ of each other" is not an equivalence relation.)

Lemma 2 can be modified to prove that the regret of the algorithm is bounded by

$$2\epsilon T + \sum_{\substack{1\leq i<j\leq n,\\ \Delta_{i,j}>\epsilon}} \frac{8}{\Delta_{i,j}^2}\left(\Delta_{i,i+1}+\Delta_{j-1,j}\right).$$

This can be seen by rewriting Equation (5) as

$$\mathbb{E}\left[\sum_{t=1}^{T}\sum_{j=2}^{n}\mathbf{1}\left\{\mathcal{E}^{(1)}_{i_t^*,j}(t)\right\}\sum_{i=i_t^*}^{i_\epsilon(j)-1}\Delta_{i,i+1}\right] + \mathbb{E}\left[\sum_{t=1}^{T}\sum_{j=2}^{n}\mathbf{1}\left\{\mathcal{E}^{(1)}_{i_t^*,j}(t)\right\}\sum_{i=i_\epsilon(j)}^{j-1}\Delta_{i,i+1}\right]$$

and noting that the second term is at most

$$\mathbb{E}\left[\sum_{t=1}^{T}\sum_{j=2}^{n}\mathbf{1}\left\{\mathcal{E}^{(1)}_{i_t^*,j}(t)\right\}\epsilon\right] = \mathbb{E}\left[\epsilon\sum_{t=1}^{T}1\right] = \epsilon T,$$

since only one of the events $\mathcal{E}^{(1)}_{i_t^*,j}(t)$ (corresponding to $j = x_t$) can occur for each t. Equation (6) can be similarly modified by splitting the summation $j = i+1\ldots x_t$ to $j = i+1\ldots j_\epsilon(i)$ and $j = j_\epsilon(i)+1\ldots x_t$.

Similarly, Lemma 3 can be modified as follows. In equation (7), instead of rewriting $\sum_{j:j>i}\Delta_{i,j}^{-2}$, we rewrite

$$\sum_{j:j>i,i<i_\epsilon(j)}\Delta_{i,j}^{-2}$$

to get

$$2\int_{y=0}^{\infty}\#\left\{j > i, \epsilon < \Delta_{i,j} \leq y\right\} y^{-3}\mathrm{d}y,$$

in Equation (8).

Equation (9) can be rewritten as

$$\sum_{j=1}^{n}\sum_{i=1}^{i_\epsilon(j)-1}\Delta_{i,j}^{-2}\Delta_{i,i+1}.$$

The rest of the analysis goes through as it is written, except that the limits of integration in Equation (10) now become $y = \max\{\epsilon, \Delta_{j-1,j}\}\ldots\infty$ instead of $y = \Delta_{j-1,j}\ldots\infty$, resulting in the final expression of

$$2\sum_{j=2}^{n}\left(\max\{\epsilon, \Delta_{j-1,j}\}\right)^{-1},$$

in Equation (11).

Therefore, the denominators of regret expression in Theorem 1 can be made at least ϵ, if we are willing to pay $2\epsilon T$ upfront in terms of regret.

3.1.2 Lower bound

In this section, assuming that the means μ_i are bounded away from 0 and 1, we prove that in terms of the regret, the FTAL algorithm presented in the section above is optimal (up to constant factors). This is done by showing the following lower bound on the regret guarantee of any algorithm.

Lemma 5 *Assume that the means μ_i are bounded away from 0 and 1. Any algorithm for the stochastic version of the best expert problem must have regret at least*

$$\Omega\left(\sum_{i=1}^{n-1}\frac{1}{\Delta_{i,i+1}}\right),$$

as T becomes large enough.

To prove this lemma, we first prove its special case for the case of two experts.

Lemma 6 *Suppose we are given two numbers $\mu_1 > \mu_2$, both lying in an interval $[a, b]$ such that $0 < a < b < 1$, and suppose we are given any online algorithm ϕ for the best expert problem with two experts. Then there is an input instance in the stochastic rewards model, with two experts L and R whose payoff distributions are Bernoulli random variables with means μ_1 and μ_2 or vice-versa, such that for large enough T, the regret of algorithm ϕ is*

$$\Omega\left(\delta^{-1}\right),$$

where $\delta = \mu_1 - \mu_2$ and the constants inside the $\Omega(\cdot)$ may depend on a, b.

Proof: Let us define some joint distributions: p is the distribution in which both experts have average payoff μ_1, q_L is the distribution in which they have payoffs (μ_1, μ_2) (left is better), and q_R is the distribution in which they have payoffs (μ_2, μ_1) (right expert is better).

Let us define the following events: E_t^L is true if ϕ picks L at time t, and similarly E_t^R.

We denote by $p^t(\cdot)$ the joint distribution for first t time steps, where the distribution of rewards in each time period is $p(\cdot)$. Similarly for $q^t(\cdot)$. We have $p^t[E_t^L] + p^t[E_t^R] = 1$. Therefore, for every t, there exists $M \in \{L, R\}$ such that $p^t[E_t^M] \geq 1/2$. Similarly, there exists $M \in \{L, R\}$ such that

$$\#\left\{t : 1 \leq t \leq T, \quad p^t[E_t^M] \geq \frac{1}{2}\right\} \geq \frac{T}{2}.$$

Take $T_0 = \frac{c}{\delta^2}$ for a small enough constant c. We will prove the claim below for $T = T_0$; for larger values of T, the claim follows easily from this.

Without loss of generality, assume that $M = L$. Now assume the algorithm faces the input distribution q_R, and define $q = q_R$. Using $\mathsf{KL}(\cdot;\cdot)$ to denote the KL-divergence of two distributions, we have

$$\begin{aligned}\mathsf{KL}(p^t; q^t) &\leq \mathsf{KL}(p^T; q^T) = T \cdot \mathsf{KL}(p; q)\\ &= c\delta^{-2} \cdot \mathsf{KL}(\mu_1; \mu_2) \leq c\delta^{-2} \cdot \mathcal{O}(\delta^2) \leq \frac{1}{50},\end{aligned}$$

for a small enough value of c which depends on a and b because the constant inside the $\mathcal{O}(\cdot)$ in the line above depends on a and b.

Karp and Kleinberg [KK07] prove the following lemma. If there is an event E with $p(E) \geq 1/3$ and $q(E) < 1/3$, then

$$\mathsf{KL}(p; q) \geq \frac{1}{3} \ln\left(\frac{1}{3q(E)}\right) - \frac{1}{e}. \tag{12}$$

We have that for at least $T/2$ values of t, $p^t(E_t^L) \geq 1/3$ (it is actually at least $1/2$). In such time steps, we either have $q^t(E_t^L) \geq 1/3$ or the lemma applies, yielding

$$\frac{1}{50} \geq \mathsf{KL}(p^t; q^t) \geq \frac{1}{3} \ln\left(\frac{1}{q^t(E_t^L)}\right) - \frac{1}{e}.$$

This gives

$$q^t(E_t^L) \geq \frac{1}{10}.$$

Therefore, the regret of the algorithm in time period t is at least

$$\mu_1 - \left(\frac{9}{10}\mu_1 + \frac{1}{10}\mu_2\right) \geq \frac{1}{10}\delta.$$

Since $T = \Omega(\delta^{-2})$, we have that the regret is at least

$$\frac{1}{10}\delta \cdot \Omega(\delta^{-2}) = \Omega(\delta^{-1}).$$

This finishes the proof of the lower bound for two experts. ■

Proof of Lemma 5: Let us group experts in pairs of 2 as $(2i-1, 2i)$ for $i = 1, 2, \ldots, \lfloor n/2 \rfloor$. Apply the two-expert lower bound from Lemma 6 by creating a series of time steps when $A_t = \{2i-1, 2i\}$ for each i. (We need a sufficiently large time horizon — namely $T \geq \sum_{i=1}^{\lfloor n/2 \rfloor} c\Delta_{2i-1,2i}^{-2}$ — in order to apply the lower bound to all $\lfloor n/2 \rfloor$ two-expert instances.) The total regret suffered by any algorithm is the sum of regret suffered in the independent $\lfloor n/2 \rfloor$ instances defined above. Using the lower bound from Lemma 6, we get that the regret suffered by any algorithm is at least

$$\sum_{i=1}^{\lfloor n/2 \rfloor} \Omega\left(\frac{1}{\Delta_{2i-1,2i}}\right).$$

Similarly, if we group the experts in pairs according to $(2i, 2i+1)$ for $i = 1, 2, \ldots, \lfloor n/2 \rfloor$, then we get a lower bound of

$$\sum_{i=1}^{\lfloor n/2 \rfloor} \Omega\left(\frac{1}{\Delta_{2i,2i+1}}\right).$$

Since both of these are lower bounds, so is their average, which is

$$\frac{1}{2}\sum_{i=1}^{n-1} \Omega\left(\frac{1}{\Delta_{i,i+1}}\right) = \Omega\left(\sum_{i=1}^{n-1} \Delta_{i,i+1}^{-1}\right).$$

This proves the lemma. ■

3.2 Multi-armed bandit setting

We now turn our attention to the multi-armed bandit setting against a stochastic adversary. We first present a variant of UCB1 algorithm [ACBF02], and then present a matching lower bound based on idea from Lai and Robbins [LR85], which is a constant factor away from the UCB1-like upper bound.

3.2.1 Upper bound (algorithm: AUER)

Here the optimal algorithm is again a natural extension of the UCB1 algorithm [ACBF02] to the sleeping-bandits case. In a nutshell, the algorithm keeps track of the running average of payoffs received from each arm, and also a confidence interval of width $2\sqrt{\frac{8\ln t}{n_{j,t}}}$ around arm j, where t is the current time interval and $n_{j,t}$ is the number of times j's payoff has been observed (number of times arm j has been played). At time t, if an arm becomes available for the first time then the algorithm chooses it. Otherwise the algorithm optimistically picks the arm with highest "upper estimated reward" (or "upper confidence bound" in UCB1 terminology) among the available arms. That is, it picks the arm $j \in A_t$ with maximum $\hat{\mu}_{j,t} + \sqrt{\frac{8 \ln t}{n_{j,t}}}$ where $\hat{\mu}_{j,t}$ is the mean of the observed rewards of arm j up to time t. The algorithm is shown in Figure 2. The algorithm is called **A**wake **U**pper **E**stimated **R**eward (AUER).

```
Initialize $z_i = 0$ and $n_i = 0$ for all $i \in [n]$.
for $t = 1$ to $T$ do
    if $\exists j \in A_t$ s.t. $n_j = 0$ then
        Play arm $x(t) = j$
    else
        Play arm
        $x(t) = \arg\max_{i \in A_t} \left(\frac{z_i}{n_i} + \sqrt{\frac{8 \log t}{n_i}}\right)$
    end
    Observe payoff $r_{x(t)}(t)$ for arm $x(t)$
    $z_{x(t)} \leftarrow z_{x(t)} + r_{x(t)}(t)$
    $n_{x(t)} \leftarrow n_{x(t)} + 1$
end
```

Algorithm 2: The AUER algorithm for sleeping bandit problem with stochastic adversary.

We first need to state a claim about the confidence intervals that we are using.

Lemma 7 *With the definition of $n_{i,t}$ and μ_i and $\hat{\mu}_i$, the following holds for all $1 \le i \le n$ and $1 \le t \le T$:*

$$\mathbb{P}\left[\mu_i \notin \left[\hat{\mu}_{i,t} - \sqrt{\frac{8 \ln t}{n_{i,t}}}, \hat{\mu}_{i,t} + \sqrt{\frac{8 \ln t}{n_{i,t}}}\right]\right] \le \frac{1}{t^4}.$$

Proof: The proof is an application of Chernoff-Hoeffding bounds, and follows from [ACBF02, pp. 242–243]. ■

Theorem 8 *The regret of the* AUER *algorithm is at most*

$$(64 \ln T) \cdot \sum_{j=1}^{n-1} \frac{1}{\Delta_{j,j+1}}.$$

up to time T.

The theorem follows immediately from the following lemma and Lemma 3.

Lemma 9 *The* AUER *algorithm has a regret of at most*

$$(32 \ln T) \cdot \sum_{j=2}^{n} \sum_{i=1}^{j-1} \left(\frac{1}{\Delta_{i,j}^2}\right) \Delta_{i,i+1}$$

Proof: We bound the regret of the algorithm arm by arm. Let us consider an arm $2 \le j \le n$. Let us count the number of times j was played, where some arm in $1, 2, \ldots, i$ could have been played (in these iterations, the regret accumulated is at least $\Delta_{i,j}$ and at most $\Delta_{1,j}$). Call this $N_{i,j}$ for $i < j$. We claim that $N_{i,j} \le \frac{32 \ln T}{\Delta_{i,j}^2}$ with probability $1 - \frac{2}{t^4}$.

Let us define $Q_{i,j} = \frac{32 \ln T}{\Delta_{i,j}^2}$. We want to claim that after playing j for $Q_{i,j}$ number of times, we will not make the mistake of choosing j instead of something from the set $\{1, 2, \ldots, i\}$; that is, if some arm in $[i]$ is awake as well as j is awake, then some awake arm in $[i]$ will be chosen, and not the arm j (with probability at least $1 - \frac{2}{t^4}$).

Let us bound the probability of choosing j when $A_t \cap [i] \neq \emptyset$ after j has been played $Q_{i,j}$ number of times.

$$\sum_{t=Q_{i,j}+1}^{T} \sum_{k=Q_{i,j}+1}^{T} \mathbb{P}\Big[(x_t = j) \wedge (j \text{ is played } k\text{-th time}) \wedge (A_t \cap [i] \neq \emptyset)\Big]$$

$$\le \sum_{t=Q_{i,j}+1}^{T} \sum_{k=Q_{i,j}+1}^{T} \mathbb{P}\left[(n_{j,t} = k) \wedge \left(\hat{\mu}_{j,t} + \sqrt{\frac{8 \ln t}{k}} \ge \hat{\mu}_{h_t,t} + \sqrt{\frac{8 \ln t}{n_{h_t,t}}}\right)\right],$$

where h_t is the index g in $A_t \cap [i]$ which maximizes $\hat{\mu}_{g,t} + \sqrt{(8 \ln t)/n_{g,t}}$, i.e. $h = \arg\max_{g \in A_t} \hat{\mu}_{g,t} + \sqrt{(8 \ln t)/n_{g,t}}$

$$= \sum_{t=Q_{i,j}+1}^{T} \sum_{k=Q_{i,j}+1}^{T} \mathcal{O}\left(\frac{1}{t^4}\right) + \mathbb{P}\left[\mu_j + \Delta_{i,j} \ge \mu_{h_t}\right]$$

$$= \mathcal{O}(1).$$

Here, the first $\frac{1}{t^4}$ term comes from the probability that j's confidence interval might be wrong, or h_t's confidence interval might be wrong (it follows from Lemma 7). Since $k > \frac{32 \ln t}{\Delta_{i,j}^2}$, j's confidence interval is at most $\Delta_{i,j}/2$ wide. Therefore, with probability $1 - \frac{2}{t^4}$, we have $\hat{\mu}_{j,t} + \sqrt{\frac{8 \ln t}{k}} \le \mu_j + \Delta_{i,j}$ and $\hat{\mu}_{h_t,t} + \sqrt{\frac{8 \ln t}{n_{h_t,t}}} \ge \mu_{h_t}$. Also, the probability $\mathbb{P}[\mu_j + \Delta_{i,j} \ge \mu_{h_t}] = 0$ since we know that $\mu_j + \Delta_{i,j} \le \mu_{h_t}$ as $h_t \in [i]$. Therefore, we can mess up only constant number of times between $[i]$ and j after j has been played $Q_{i,j}$ number of times. We get that

$$\mathbb{E}[N_{i,j}] \le Q_{i,j} + \mathcal{O}(1).$$

Now, it is easy to bound the total regret of the algorithm, which is

$$\mathbb{E}\left[\sum_{j=2}^{n} \sum_{i=1}^{j-1} (N_{i,j} - N_{i-1,j}) \Delta_{i,j}\right] \tag{13}$$

$$= \sum_{j=2}^{n} \sum_{i=1}^{j-1} N_{i,j} \left(\Delta_{i,j} - \Delta_{i+1,j}\right),$$

which follows by regrouping of terms and the convention that $N_{0,j} = 0$ and $\Delta_{j,j} = 0$ for all j. Taking the expectation of this gives the regret bound of

$$(32 \ln T) \cdot \sum_{j=2}^{n} \sum_{i=1}^{j-1} \left(\frac{1}{\Delta_{i,j}^2}\right) (\Delta_{i,j} - \Delta_{i+1,j}).$$

This gives the statement of the lemma. ■

Remarks for small $\Delta_{i,i+1}$ As noted in the case of expert setting, the upper bound above become trivial if some $\Delta_{i,i+1}$ are small. In such case, the proof can be modified by changing equation (13) as follows.

$$\begin{aligned}
&\sum_{j=2}^{n}\sum_{i=1}^{j-1}(N_{i,j}-N_{i-1,j})\Delta_{i,j}\\
&=\sum_{j=2}^{n}\sum_{i=1}^{i_\epsilon(j)}(N_{i,j}-N_{i-1,j})\Delta_{i,j}\\
&\quad+\sum_{j=2}^{n}\sum_{i=i_\epsilon(j)+1}^{j-1}(N_{i,j}-N_{i-1,j})\Delta_{i,j}\\
&\le\sum_{j=2}^{n}\sum_{i=1}^{i_\epsilon(j)-1}N_{i,j}\Delta_{i,i+1}+\sum_{j=2}^{n}N_{i_\epsilon(j),j}\Delta_{i_\epsilon(j),j}\\
&\quad+\sum_{j=2}^{n}\sum_{i=i_\epsilon(j)+1}^{j-1}(N_{i,j}-N_{i-1,j})\epsilon\\
&\le\sum_{j=2}^{n}\sum_{i=1}^{i_\epsilon(j)-1}N_{i,j}\Delta_{i,i+1}+\epsilon\sum_{j=2}^{n}N_{i_\epsilon(j),j}\\
&\quad+\epsilon\sum_{j=2}^{n}(N_{j-1,j}-N_{i_\epsilon(j),j})\\
&\le\sum_{1\le i<j\le n,\Delta_{i,j}>\epsilon}N_{i,j}\Delta_{i,i+1}+\epsilon T,
\end{aligned}$$

where the last step follows from $\sum_{j=2}^{n}N_{j-1,j}\le T$.

Taking the expectation, and using the modification of Lemma 3 suggested in Section 3.1.1 gives us an upper bound of

$$\epsilon T+(64\ln T)\sum_{i=1}^{n-1}(\max\{\epsilon,\Delta_{i,i+1}\})^{-1},$$

for any $\epsilon\ge 0$.

3.2.2 Lower bound

In this section, we prove that the AUER algorithm presented is information theoretically optimal up to constant factors when the means of arms μ_i's are bounded away from 0 and 1. We do this by presenting a lower bound of

$$\Omega\left(\ln T\cdot\sum_{i=1}^{n-1}\Delta_{i,i+1}^{-1}\right)$$

for this problem. This is done by closely following the lower bound of Lai and Robbins [LR85] for two armed bandit problems. The difference is that Lai and Robbins prove their lower bound only in the case when T approaches ∞, but we want to get bounds that hold for finite T. Our main result is stated in the following lemma.

Lemma 10 *Suppose there are n arms and n Bernoulli distributions P_i with means μ_i, with each $\mu_i\in[\alpha,\beta]$ for some $0<\alpha<\beta<1$. Let ϕ be an algorithm for picking among n arms which, up to time t, plays a suboptimal bandit at most $o(t^a)$ number of times for every $a>0$. Then, there is an input instance with n arms endowed with some permutation of above mentioned n distributions, such that the regret of ϕ has to be at least*

$$\Omega\left(\sum_{i=1}^{n-1}\frac{(\log t)(\mu_i-\mu_{i+1})}{\mathsf{KL}(\mu_{i+1};\mu_i)}\right),$$

for $t\ge n^2$.

We first prove the result for two arms. For this, in the following, we extend the Lai and Robbins result so that it holds (with somewhat worse constants) for finite T, rather than only in the limit $T\to\infty$.

Lemma 11 *Let there be two arms and two distributions $P_1(\cdot)$ and $P_2(\cdot)$ with means μ_1 and μ_2 with $\mu_i\in[\alpha,\beta]$ for $i=1,2$ and $0<\alpha<\beta<1$. Let ϕ be any algorithm for choosing the arms which never picks the worse arm (for any values of μ_1 and μ_2 in $[\alpha,\beta]$) more than $o(T^a)$ times (for any value of $a>0$).*

Then there exists an instance for ϕ with two arms endowed with two distributions above (in some order) such that the regret of the algorithm if presented with this instance is at least

$$\Omega\left(\frac{(\log t)(\mu_1-\mu_2)}{\mathsf{KL}(\mu_2;\mu_1)}\right),$$

where the constant inside the big-omega is at least $1/2$.

Proof: Since we are proving a lower bound, we just focus on Bernoulli distributions, and prove that if we have two bandits, with Bernoulli payoffs with means μ_1 and μ_2 such that $\alpha\le\mu_2<\mu_1\le\beta$, then we can get the above mentioned lower bound.

Let us fix a $\delta<1/10$. From the assumption that μ_1 and μ_2 are bounded away from 0 and 1, there exists a Bernoulli distribution with mean $\lambda>\mu_1$ with

$$|\,\mathsf{KL}(\mu_2;\lambda)-\mathsf{KL}(\mu_2;\mu_1)|\le\delta\cdot\mathsf{KL}(\mu_2;\mu_1),$$

because of the continuity of KL divergence in its second argument.

This claim provides us with a Bernoulli distribution with mean λ and

$$\mathsf{KL}(\mu_2;\lambda)\le(1+\delta)\,\mathsf{KL}(\mu_2;\mu_1). \tag{14}$$

From now on, until the end of the proof, we work with the following two distributions on t-step histories: p is the distribution induced by Bernoulli arms with means (μ_1,μ_2), and q is the distribution induced by Bernoulli arms with means (μ_1,λ). From the assumption of the lemma, we have

$$\mathbb{E}_q[t-n_{2,t}]\le o(t^a),\qquad\text{for all } a>0.$$

We choose any $a<\delta$. By an application of Markov's inequality, we get that

$$\begin{aligned}
&\mathbb{P}_q[n_{2,t}<(1-\delta)(\log t)/\,\mathsf{KL}(\mu_2;\lambda)]\\
&\quad\le\frac{\mathbb{E}_q[t-n_{2,t}]}{t-(1-\delta)(\log t)/\,\mathsf{KL}(\mu_2;\lambda)}\le o(t^{a-1}).
\end{aligned}\tag{15}$$

Let $\mathcal{E}$ denote the event that $n_{2,t} < (1-\delta)\log t/\,\mathsf{KL}(\mu_2;\lambda)$. If $\mathbb{P}_p(\mathcal{E}) < 1/3$, then

$$\begin{aligned}\mathbb{E}_p[n_{2,t}] &\geq \mathbb{P}_p(\overline{\mathcal{E}}) \cdot (1-\delta)\log t/\,\mathsf{KL}(\mu_2,\lambda) \\ &\geq \frac{2}{3} \cdot (1-\delta)\log t/\,\mathsf{KL}(\mu_2,\lambda) \\ &\geq \frac{2}{3}\left(\frac{1-\delta}{1+\delta}\frac{\log t}{\mathsf{KL}(\mu_2;\mu_1)}\right),\end{aligned}$$

which implies the stated lower bound for $\delta = 1/10$.

Henceforth, we will assume $\mathbb{P}_p(\mathcal{E}) \geq 1/3$. We have $\mathbb{P}_q(\mathcal{E}) < 1/3$ using (15). Now we can apply the lemma from [KK07] stated in (12), we have

$$\begin{aligned}\mathsf{KL}(p;q) &\geq \frac{1}{3}\ln\left(\frac{1}{3\,o(t^{a-1})}\right) - \frac{1}{e} \\ &= (1-a)\ln t - \mathcal{O}(1). \qquad (16)\end{aligned}$$

The chain rule for KL divergence [CT99, Theorem 2.5.3] implies

$$\mathsf{KL}(p;q) = \mathbb{E}_p[n_{2,t}] \cdot \mathsf{KL}(\mu_2;\lambda) \qquad (17)$$

Combining (16) with (17), we get

$$\begin{aligned}\mathbb{E}_{\mu_1,\mu_2}[n_{2,t}] &\geq \frac{(1-a)\ln t - \mathcal{O}(1)}{\mathsf{KL}(\mu_2;\lambda)} \\ &\geq \frac{1-a}{1+\delta}\frac{\ln t}{\mathsf{KL}(\mu_2;\mu_1)} - \mathcal{O}(1). \qquad (18)\end{aligned}$$

Using $a < \delta < 1/10$, the regret bound follows. ■

We now extend the result from 2 to n bandits.

Proof of Lemma 10: A naive way to extend the lower bound is to divide the time line between $n/2$ blocks of length $2T/n$ each and use $n/2$ separate two-armed bandit lower bounded as done in the proof of Lemma 5.

We can pair the arms in pairs of $(2i-1, 2i)$ for $i = 1, 2, \ldots, \lfloor n/2 \rfloor$. We present the algorithm with two arms $2i-1$ and $2i$ in the i-th block of time. The lower bound then is

$$\begin{aligned}&\log\left(\frac{T}{n}\right)\left(\frac{\mu_1-\mu_2}{\mathsf{KL}(\mu_2;\mu_1)} + \cdots + \frac{\mu_{2\lfloor n/2\rfloor-1} - \mu_{2\lfloor n/2\rfloor}}{\mathsf{KL}(\mu_{2\lfloor n/2\rfloor};\mu_{2\lfloor n/2\rfloor-1})}\right) \\ &\quad = \Omega\left((\log T)\cdot\left(\sum_{i=1}^{\lfloor n/2\rfloor}\Delta_{2i,2i-1}^{-1}\right)\right),\end{aligned}$$

if we take $T > n^2$. Using the fact that $\mu_i \in [\alpha, \beta]$, we have $\mathsf{KL}(\mu_i;\mu_j) = \mathcal{O}(\Delta_{i,j}^{-2})$ which justifies the derivation of the second line above.

We get a similar lower bound by presenting the algorithm with $(2i, 2i+1)$, which gives us a lower bound of

$$\Omega\left((\log T)\cdot\left(\sum_{i=1}^{\lfloor n/2\rfloor}\Delta_{2i,2i+1}^{-1}\right)\right).$$

Taking their averages gives the required lower bound, proving the lemma. ■

4 Adversarial Model of Rewards

We now turn our attention to the case where no distributional assumptions are made on the generation of rewards. In this section we prove information theoretic lower bounds on the regret of any online learning algorithm for both the best expert and the multi-armed bandit settings. We also present online algorithms whose regret is within a constant factor of the lower bound for the expert setting and within a sublogarithmic factor of the lower bound for the bandit setting. Unlike in the stochastic rewards setting, however, these algorithms are not computationally efficient. It is an open problem if there exists an efficient algorithm whose regret grows as polynomial in n.

4.1 Best expert

Theorem 12 *For every online algorithm* ALG *and every time horizon T, there is an adversary such that the algorithm's regret with respect to the best ordering, at time T, is*

$$\Omega(\sqrt{Tn\log(n)}).$$

Proof: We construct a randomized oblivious adversary (i.e., a distribution on input sequences) such that the regret of any algorithm ALG is at least $\Omega(\sqrt{Tn\log(n)})$. The adversary partitions the timeline $\{1, 2, \ldots, T\}$ into a series of *two-expert games*, i.e. intervals of consecutive rounds during which only two experts are awake and all the rest are asleep. In total there will be $Q(n) = \Theta(n\log n)$ two-expert games, where $Q(n)$ is a function to be specified later in (20). For $i = 1, 2, \ldots, Q(n)$, the set of awake experts throughout the i-th two-experts game is a pair $A^{(i)} = \{x_i, y_i\}$, determined by the adversary based on the (random) outcomes of previous two-experts games. The precise rule for determining the elements of $A^{(i)}$ will be explained later in the proof.

Each two-experts game runs for $T_0 = T/Q(n)$ rounds, and the payoff functions for the rounds are independent, random bijections from $A^{(i)}$ to $\{0, 1\}$. Letting $g^{(i)}(x_i)$, $g^{(i)}(y_i)$ denote the payoffs of x_i and y_i, respectively, during the two-experts game, it follows from Khintchine's inequality [Khi23] that

$$\mathbb{E}\left(\left|g^{(i)}(x_i) - g^{(i)}(y_i)\right|\right) = \Omega\left(\sqrt{T_0}\right). \qquad (19)$$

The expected payoff for any algorithm can be at most $\frac{T_0}{2}$, so for each two-experts game the regret of any algorithm is at least $\Omega(\sqrt{T_0})$. For each two-experts game we define the *winner* W_i to be the element of $\{x_i, y_i\}$ with the higher payoff in the two-experts game; we will adopt the convention that $W_i = x_i$ in case of a tie. The *loser* L_i is the element of $\{x_i, y_i\}$ which is not the winner.

The adversary recursively constructs a sequence of $Q(n)$ two-experts games and an ordering of the experts such that the winner of every two-experts game precedes the loser in this ordering. (We call such an ordering *consistent* with the sequence of games.) In describing the construction, we assume for convenience that n is a power of 2. If $n = 2$ then we set $Q(2) = 1$ and we have a single two-experts game and an ordering in which the winner precedes the loser. If $n > 2$ then we recursively construct a sequence of games and an ordering consistent with those games, as follows:

1. We construct $Q(n/2)$ games among the experts in the set $\{1, 2, \ldots, n/2\}$ and an ordering $\prec_1$ consistent with those games.
2. We construct $Q(n/2)$ games among the experts in the set $\{(n/2) + 1, \ldots, n\}$ and an ordering $\prec_2$ consistent with those games.
3. Let $k = 2Q(n/2)$. For $i = 1, 2, \ldots, n/2$, we define x_{k+i} and y_{k+i} to be the i-th elements in the orderings $\prec_1, \prec_2$, respectively. The $(k+i)$-th two-experts game uses the set $A^{(k+i)} = \{x_{k+i}, y_{k+i}\}$.
4. The ordering of the experts puts the winner of the game between x_{k+i} and y_{k+i} before the loser, for every $i = 1, 2, \ldots, n/2$, and it puts both elements of $A^{(k+i)}$ before both elements of $A^{(k+i+1)}$.

By construction, it is clear that the ordering of experts is consistent with the games, and that the number of games satisfies the recurrence

$$Q(n) = 2Q(n/2) + n/2, \tag{20}$$

whose solution is $Q(n) = \Theta(n \log n)$.

The best ordering of experts achieves a payoff at least as high as that achieved by the constructed ordering which is consistent with the games. By (19), the expected payoff of that ordering is $T/2 + Q(n) \cdot \Omega(\sqrt{T_0})$. The expected payoff of ALG in each round t is $1/2$, because the outcome of that round is independent of the outcomes of all prior rounds. Hence the expected payoff of ALG is only $T/2$, and its regret is

$$\begin{aligned} Q(n) \cdot \Omega(\sqrt{T_0}) &= \Omega(n \log n \sqrt{T/(n \log n)}) \\ &= \Omega(\sqrt{Tn \log n}). \end{aligned}$$

This proves the theorem. ■

It is interesting to note that the adversary that achieves this lower bound is not adaptive in either choosing the payoffs or choosing the awake experts at each time step. It only needs to be able to carefully coordinate which experts are awake based on the payoffs at previous time steps.

Even more interesting, this lower bound is tight, so an adaptive adversary is not more powerful than an oblivious one. There is a learning algorithm that achieves a regret of $O(\sqrt{Tn \log(n)})$, albeit not computationally efficient. To achieve this regret we transform the sleeping experts problem to a problem with $n!$ experts that are always awake. In the new problem, we have one expert for each ordering of the original n experts. At each round, each of the $n!$ experts makes the same prediction as the highest ranked expert in its corresponding ordering, and receives the payoff of that expert.

Theorem 13 *An algorithm that makes predictions using* Hedge *on the transformed problem achieves $O(\sqrt{Tn \log(n)})$ regret with respect to the best ordering.*

Proof: Every expert in the transformed problem receives the payoff of its corresponding ordering in the original problem. Since Hedge achieves regret $O(\sqrt{T \log(n!)})$ with respect to the best expert in the transformed problem, the same regret is achieved by the algorithm in the original problem. ■

4.2 Multi-armed bandit setting

Theorem 14 *For every online algorithm* ALG *and every time horizon T, there is an adversary such that the algorithm's regret with respect to the best ordering, at time T, is $\Omega(n\sqrt{T})$.*

Proof: To prove the lower bound we will rely on the lower bound proof for the multi-armed bandit in the usual setting when all the experts are awake [ACBFS02]. In the usual bandit setting with a time horizon of T_0, any algorithm will have at least $\Omega(\sqrt{T_0 n})$ regret with respect to the best expert. To ensure this regret, the input sequence is generated by sampling T_0 times independently from a distribution in which every bandit but one receives a payoff of 1 with probability $\frac{1}{2}$ and 0 otherwise. The remaining bandit, which is chosen at random, incurs a payoff of 1 with probability $\frac{1}{2} + \epsilon$ for an appropriate choice of ϵ.

To obtain the lower bound for the sleeping bandits setting we set up a sequence of n multi-armed bandit games as described above. Each game will run for $T_0 = \frac{T}{n}$ rounds. The bandit that received the highest payoff during the game will become asleep and unavailable in the rest of the games.

In game i, any algorithm will have a regret of at least $\Omega\left(\sqrt{\frac{T}{n}(n-i)}\right)$ with respect to the best bandit in that game. In consequence, the total regret of any learning algorithm with respect to the best ordering is:

$$\begin{aligned} &\sum_{i=1}^{n-1} \sqrt{\frac{T}{n}(n-i)} = \sqrt{\frac{T}{n}} \sum_{j=1}^{n-1} j^{1/2} \\ &\geq \sqrt{\frac{T}{n}} \int_{x=0}^{n-1} x^{1/2} dx = \sqrt{\frac{T}{n}} \frac{2}{3} \left((n-1)^{3/2}\right) \\ &= \Omega\left(n\sqrt{T}\right). \end{aligned}$$

The theorem follows. ■

To get an upper bound on regret, we will use the Exp4 algorithm [ACBFS02]. Since Exp4 requires an oblivious adversary, in the following, we assume that the adversary is oblivious (as opposed to adaptive). Exp4 chooses an action by combining the advice of a set of "experts." At each round, each expert provides advice in the form of a probability distribution over actions. In particular the advice can be a point distribution concentrated on a single action. (It is required that at least one of the experts is the *uniform expert* whose advice is always the uniform distribution over actions.) To use Exp4 for the sleeping experts setting, in addition to the uniform expert we have an expert for each ordering over actions. At each round, the advice of that expert is a point distribution concentrated on the highest ranked action in the corresponding ordering.

Since the uniform expert may advise us to pick actions which are not awake, we assume for convenience that the problem is modified as follows. Instead of being restricted to choose an action in the set A_t at time t, the algorithm is allowed to choose any action at all, with the proviso that the payoff of an action in the complement of A_t is defined to be 0. Note that any algorithm for this modified problem can easily be transformed into an algorithm for the original problem: every time the algorithm chooses an action in the complement of A_t we instead play an arbitrary action in A_t. Such a transformation can only increase the algorithm's payoff, i.e. decrease the regret. Hence, to prove the regret bound asserted in Theorem 15 below, it suffices to prove the same bound for the modified problem.

Theorem 15 *Against an oblivious adversary, the* Exp4 *algorithm as described above achieves a regret of $O(n\sqrt{T\log(n)})$ with respect to the best ordering.*

Proof: We have n actions and $1+n!$ experts, so the regret of Exp4 with respect to the payoff of the best expert is $\mathcal{O}(\sqrt{Tn\log(n!+1)})$ [ACBFS02]. Since the payoff of each expert is exactly the payoff of its corresponding ordering we obtain the statement of the theorem. ■

The upper bound and lower bound differ by a factor of $\mathcal{O}(\sqrt{\log(n)})$. The same gap exists in the usual multi-armed bandit setting where all actions are available at all times, hence closing the logarithmic gap between the lower and upper bounds in Theorems 14 and 15 is likely to be as difficult as closing the corresponding gap for the nonstochastic multi-armed bandit problem itself.

5 Conclusions

We have analyzed algorithms for full-information and partial-information prediction problems in the "sleeping experts" setting, using a novel benchmark which compares the algorithm's payoff against the best payoff obtainable by selecting available actions using a fixed total ordering of the actions. We have presented algorithms whose regret is information-theoretically optimal in both the stochastic and adversarial cases. In the stochastic case, our algorithms are simple and computationally efficient. In the adversarial case, the most important open question is whether there is a computationally efficient algorithm which matches (or nearly matches) the regret bounds achieved by the exponential-time algorithms presented here.

References

[ACBF02] Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2-3):235–256, 2002.

[ACBFS02] Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM J. Comput.*, 32(1):48–77, 2002.

[Azu67] K. Azuma. Weighted sums of certain dependent random variables. *Tohoku Math. J.*, 19:357–367, 1967.

[BM05] Avrim Blum and Yishay Mansour. From external to internal regret. In *COLT*, pages 621–636, 2005.

[CBFH$^+$97] Nicolò Cesa-Bianchi, Yoav Freund, David Haussler, David P. Helmbold, Robert E. Schapire, and Manfred K. Warmuth. How to use expert advice. *J. ACM*, 44(3):427–485, 1997.

[CT99] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. J. Wiley, 1999.

[FSSW97] Yoav Freund, Robert E. Schapire, Yoram Singer, and Manfred K. Warmuth. Using and combining predictors that specialize. In *STOC*, pages 334–343, 1997.

[Han57] J. Hannan. Approximation to Bayes risk in repeated plays. volume 3, pages 97–139, 1957. in: M. Dresher, A. Tucker, P. Wolfe (Eds.), Contributions to the Theory of Games, Princeton University Press.

[Hoe63] W. Hoeffding. Probability inequalities for sums of bounded random variables. *J. American Stat. Assoc.*, 58:13–30, 1963.

[Khi23] Aleksandr Khintchine. Über dyadische Brüche. *Math Z.*, 18:109–116, 1923.

[KK07] Richard M. Karp and Robert Kleinberg. Noisy binary search and its applications. In *SODA*, pages 881–890, 2007.

[KV05] Adam Tauman Kalai and Santosh Vempala. Efficient algorithms for on-line optimization. *J. Computer and System Sciences*, 71(3):291–307, 2005.

[LR85] T. L. Lai and Herbert Robbins. Asymptotically efficient adaptive allocations rules. *Adv. in Appl. Math.*, 6:4–22, 1985.

[LW94] Nick Littlestone and Manfred K. Warmuth. The weighted majority algorithm. *Inf. Comput.*, 108(2):212–261, 1994. An extended abstract appeared in IEEE Symposium on Foundations of Computer Science, 1989, pp. 256–261.

[LZ07] John Langford and Tong Zhang. The epoch-greedy algorithm for multiarmed bandits with side information. In *NIPS*, 2007.

[Rob] H. Robbins. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, 58:527–535.

[Vov90] V. G. Vovk. Aggregating strategies. In *COLT*, pages 371–386, 1990.

[Vov98] V. G. Vovk. A game of prediction with expert advice. *J. Comput. Syst. Sci.*, 56(2):153–173, 1998. An extended abstract appeard in COLT 1995, pp. 51–60.