Machine Learning manuscript No.
(will be inserted by the editor)

Regret Bounds for Sleeping Experts and Bandits

Robert Kleinberg · **Alexandru Niculescu-Mizil** · **Yogeshwer Sharma**

Received: date / Accepted: date

Abstract We study on-line decision problems where the set of actions that are available to the decision algorithm varies 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 adversarial rewards models. 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.

Keywords Online algorithms · Computational learning theory · Regret

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.

Robert Kleinberg was supported by NSF awards CCF-0643934, IIS-0905467, and AF-0910940, a Microsoft Research New Faculty Fellowship, and an Alfred P. Sloan Foundation Fellowship.
Alexandru Niculescu-Mizil was supported by NSF awards 0347318, 0412930 , 0427914, and 0612031.
Yogeshwer Sharma was supported by NSF grant CCF-0514628.

Robert Kleinberg
Department of Computer Science, Cornell University, Ithaca, NY 14853
E-mail: rdk@cs.cornell.edu

Alexandru Niculescu-Mizil
Department of Computer Science, Cornell University, Ithaca, NY 14853
E-mail: alexn@cs.cornell.edu
Mathematical Sciences Department, IBM T.J. Watson Research Center, Yorktown Heights, NY 10598

Yogeshwer Sharma
Department of Computer Science, Cornell University, Ithaca, NY 14853
E-mail: yogi@cs.cornell.edu

At the end of each round the algorithm receives 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.

In the prior-free setting (as is the case in this paper), the performance of the algorithm is typically measured in terms of *regret*. (See (Gittins, 1979), (Gittins & Jones, 1979) for maximization of expected reward in the Bayesian setting.) 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 gave multi-armed bandit problems their name, 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 in an adversarial way 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 1.1 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 respect to the best ordering for both the best expert setting and the multi-armed bandit setting. We first explore the case of a stochastic adversary, where the payoffs received by action 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. (Note that in this paper, the choice of which actions are available to be picked in each round is always adversarial. In other words, there is no distributional assumption on the subset of available actions.) 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[1]). 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. (2002a), 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" (Hannan, 1957; Kalai & Vempala, 2005), 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 in stochastic multi-armed bandit setting, we must modify the classic asymptotic lower bound proof of Lai and Robbins (Lai & Robbins, 1985) to obtain a bound which holds at all sufficiently large finite times. For the stochastic best expert setting, we adapt standard KL-divergence arguments to prove a precise lower bound that also holds for sufficiently large finite times. Our lower bounds in Lemma 8 and Lemma 14 don't refer to the "sleeping" version of the problem, and concern the classical best-expert setting and multi-armed bandit setting (all actions available), which might be of interest outside the context of this paper.

To prove that our lower and upper bounds are within a constant factor of each other we use a novel lemma (Lemma 4) 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 (and similarly in the best expert setting).

Next we explore the fully adversarial case where we make no assumptions on how the payoffs for each action are generated (in particular, they could depend on the time horizon T). This model has been extensively studied in both the best expert setting and the multi-armed bandit setting (see (Littlestone & Warmuth, 1994), (Auer et al., 2002b) and references therein). For the variant in which only a subset of the actions are available at any given time, we show that the regret of any learning algorithm must be at least $\Omega(\sqrt{Tn\log(n)})$ for the best expert setting and $\Omega(\sqrt{Tn^2})$ for the multi-armed bandit setting. We also present simple variants of algorithms in (Littlestone & Warmuth, 1994) and (Auer et al., 2002b) whose regret is within a constant factor of the lower bound for the best expert setting, and within $\mathcal{O}(\sqrt{\log(n)})$ of the lower bound for the multi-armed bandit setting.

The fully adversarial case, however, proves to be harder, and neither algorithm is computationally efficient. To appreciate the hardness of the fully adversarial case, we prove that, unless RP = 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 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-

[1] As is the convention in the literature, the problem instance is not allowed to depend on T in the stochastic setting. In other words, first the distributions $P_i(\cdot)$ are chosen, and then we look at regret bounds as a function of T.

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 bounded support and 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 adaptive 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. A σ-policy corresponding to the ordering σ is the policy that selects, at each time step t, the action $\sigma(A_t)$ (i.e. available action that is highest ranked by σ). The reward of a policy σ is the reward obtained by the selected action at each time step:

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

Let $r_{\max}(1:T) = \max_\sigma r_\sigma(1:T)$ ($\max_\sigma \mathbb{E}[r_\sigma(1:T)]$ in the stochastic rewards model) be the reward obtained by the best σ-policy (ordering), which is also called the benchmark. Note that in the stochastic reward model, the expectation is taken before taking the maximum over all orderings, which corresponds to the "maximum expected" reward, as opposed to the "expected maximum" reward in the adversarial setting (as is also done in the literature). We define the regret of an algorithm with respect to the best σ-policy as the expected difference between the reward obtained by the best σ-policy and the total reward of the algorithm's chosen actions $x(1), x(2), ..., x(t)$:

$$\text{regret}_x(1:T) = \mathbb{E}\left[r_{\max}(1:T) - \sum_{t=1}^{T} r_{x(t)}(t) \right], \tag{2}$$

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

1.2 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 (1994), and Vovk (1990). Cesa-Bianchi et al. (1997) further developed this paradigm in work which gave optimal regret bounds of $\sqrt{T(\ln n)}$ and Vovk (1998) characterized the achievable regret bounds in these settings.

The multi-armed bandit model was introduced by Robbins (1952). Lai and Robbins (1985) gave asymptotically optimal strategies for the stochastic version of bandit problem, where rewards for each arm are drawn from a fixed distribution in each time step.

Auer et al. (2002a) introduced the algorithm UCB1 and showed that the optimal regret bounds of $\mathcal{O}(\log T \cdot \sum_{i=1}^{n-1} \frac{1}{\mu_i - \mu_{i+1}})$ can be achieved uniformly over time for the stochastic

bandit problem (the arms are arranged such that $\mu_1 \geq \mu_2 \geq \cdots \geq \mu_n$). For the adversarial version of the multi-armed bandit problem, Auer et al. (2002b) 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 $\Omega(\sqrt{nT})$. Recently, Audibert and Bubeck (2009) proposed a $\mathcal{O}(\sqrt{Tn})$ regret algorithm for the adversarial multi-armed bandit problem closing the sub-logarithmic gap. 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 et al. (1997) and Blum and Mansour (2005) have analysed the sleeping experts problem in a different framework 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.

In contrast, our work uses a different evaluation criterion, namely the best ordering of experts. In the special case when all experts are always awake, both evaluation criteria pick the best expert. Our "best ordering" criterion can be regarded as a degenerate case (limiting case) of the "best mixture" criterion of Freund et al. as follows. For the ordering σ, we assign probabilities $\frac{1}{Z}(1, \varepsilon, \varepsilon^2, \ldots, \varepsilon^{n-1})$ to the sequence of experts $(\sigma(1), \sigma(2), \ldots, \sigma(n))$ where $Z = \frac{1-\varepsilon^n}{1-\varepsilon}$ is the normalization factor and $\varepsilon > 0$ is an arbitrarily small positive constant. The only problem is that the bounds obtained from (Freund et al., 1997) in this degenerate case are very weak. As $\varepsilon \to 0$, their bound reduces to comparing the algorithm's performance to the ordering σ's performance only for time periods when expert $\sigma(1)$ 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 (2005) 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 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

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

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 3 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 (2007) 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 $\mathscr{H}$ 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 $h \in \mathscr{H}$ 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 $\mathscr{H}$ 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 (Langford & Zhang, 2007) 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 (Langford & Zhang, 2007), we propose a very simple bandit algorithm for the stochastic setting with an explicit regret bound that is provably information-theoretically optimal.

2 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 increase in the expected 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.

2.1 Best Expert Setting

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

2.1.1 Upper Bound (Algorithm: FTAL*)*

To get an upper bound on regret we adapt the "follow the leader" algorithm (Hannan, 1957; Kalai & Vempala, 2005) 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 is more than one such expert, 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).

The performance guarantee of the algorithm FTAL is presented in the following theorem.

```
Initialize z_i = 0 and n_i = 0 for all i ∈ [n].
for t = 1 to T do
    if ∃ j ∈ A_t s.t. n_j = 0 then
        Play expert x(t) = j
    else
        Play expert x(t) = arg max_{i∈A_t} ( z_i / n_i )
    end
    Observe payoff r_i(t) for all i ∈ A_t
    z_i ← z_i + r_i(t) for all i ∈ A_t
    n_i ← n_i + 1 for all i ∈ A_t
end
```

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

Theorem 1 *Let $\Delta_{i,i+1} > 0$ for $i = 1, 2, \ldots, n-1$. Then* FTAL *algorithm has a regret of at most*

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

with respect to the best ordering.

Note that we are only considering problem instances in which different arms have different average payoffs. Also note that as $\Delta_{i,i+1}$ gets close to 0, the regret bound become vacuous. A general result will be proved in Theorem 6 which will take care of both these restrictions, and the above theorem follows as a corollary to Theorem 6 by setting $\varepsilon = 0$.

The above theorem follows immediately from the following pair of lemmas. The second of these lemmas will also be used in Section 2.2.

Lemma 2 *Let $\Delta_{i,i+1} > 0$ for $i = 1, 2, \ldots, n-1$. Then 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 (Azuma, 1967; Hoeffding, 1963) says that

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

and

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

Let us say that the FTAL algorithm suffers an (i, j)*-anomaly of type 1* at time t if $x_t = j$ and $\hat{\mu}_{j,t} - \mu_j > \Delta_{i,j}/2$; note that the definition does not require expert i to be awake at time t. Define i_t^* to be the optimal expert at time t (lowest indexed expert in A_t). 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 again that the definition does not require expert j to be awake at time t. 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 $\mathscr{E}_{i,j}^{(1)}(t)$ (resp. $\mathscr{E}_{i,j}^{(2)}(t)$), 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

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

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 $P_i(\cdot)$). Recall that $i_t^* \in A_t$ is the optimal expert at time t (i.e., the lowest-numbered element of A_t). We are now ready to bound the regret of the FTAL algorithm. A very crucial observation that we make next is that when arm i_t^* is the optimal arm in round t and arm $x_t \neq i_t^*$ is picked by the algorithm, one of the following two events must have happened: either the observed reward of arm i_t^* is much *smaller* than its actual mean $\mu_{i_t^*}$, or the observed reward of arm x_t is much *larger* than its actual mean μ_{x_t}. The first one corresponds to an (i_t^*, x_t)-anomaly of type 2, and the second one corresponds to an (i_t^*, x_t)-anomaly of type 1. We split the regret according to this classification, and bound each term in turn.

$$\begin{aligned}
\mathbb{E}\left[\sum_{t=1}^{T} \left(r_{i_t^*}(t) - r_{x_t}(t)\right)\right] &= \mathbb{E}\left[\sum_{t=1}^{T} \Delta_{i_t^*,x_t}\right] = \mathbb{E}\left[\sum_{t=1}^{T} \mathbf{1}\left\{\mathscr{E}_{i_t^*,x_t}^{(1)}(t) \vee \mathscr{E}_{i_t^*,x_t}^{(2)}(t)\right\} \Delta_{i_t^*,x_t}\right]\\
&\leq \mathbb{E}\left[\sum_{t=1}^{T} \mathbf{1}\left\{\mathscr{E}_{i_t^*,x_t}^{(1)}(t)\right\} \Delta_{i_t^*,x_t}\right] + \mathbb{E}\left[\sum_{t=1}^{T} \mathbf{1}\left\{\mathscr{E}_{i_t^*,x_t}^{(2)}(t)\right\} \Delta_{i_t^*,x_t}\right]. \quad (5)
\end{aligned}$$

With the convention that $\Delta_{i,j} = 0$ for $j \leq i$, the first term in (5) can be bounded as follows.

$$\begin{aligned}
&\mathbb{E}\left[\sum_{t=1}^{T} \mathbf{1}\left\{\mathscr{E}_{i_t^*,x_t}^{(1)}(t)\right\} \Delta_{i_t^*,x_t}\right]\\
&= \mathbb{E}\left[\sum_{t=1}^{T} \sum_{j=2}^{n} \mathbf{1}\left\{\mathscr{E}_{i_t^*,j}^{(1)}(t)\right\} \Delta_{i_t^*,j}\right] \quad \text{(Since the event } \mathscr{E}_{i_t^*,j}^{(1)}(t) \text{ occurs only for } j = x_t.\text{)}\\
&= \mathbb{E}\left[\sum_{t=1}^{T} \sum_{j=2}^{n} \mathbf{1}\left\{\mathscr{E}_{i_t^*,j}^{(1)}(t)\right\} \sum_{i=i_t^*}^{j-1} \Delta_{i,i+1}\right] \quad (6)\\
&\leq \mathbb{E}\left[\sum_{t=1}^{T} \sum_{j=2}^{n} \sum_{i=i_t^*}^{j-1} \mathbf{1}\left\{\mathscr{E}_{i,j}^{(1)}(t)\right\} \Delta_{i,i+1}\right] \quad \text{(Since } \mathbf{1}\left\{\mathscr{E}_{i_1,j}^{(1)}(t)\right\} \leq \mathbf{1}\left\{\mathscr{E}_{i_2,j}^{(1)}(t)\right\} \text{ for } i_1 \leq i_2 < j.\text{)}
\end{aligned}$$

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

Similarly, the second term in (5) can be bounded by

$$
\begin{aligned}
&\mathbb{E}\left[\sum_{t=1}^{T}\mathbf{1}\left\{\mathscr{E}_{i_t^*,x_t}^{(2)}(t)\right\}\Delta_{i_t^*,x_t}\right] \\
&= \mathbb{E}\left[\sum_{t=1}^{T}\sum_{i=1}^{n-1}\mathbf{1}\left\{\mathscr{E}_{i,x_t}^{(2)}(t)\right\}\Delta_{i,x_t}\right] \quad \text{(Since event } \mathscr{E}_{i,x_t}^{(2)}(t) \text{ occurs only for } i=i_t^*.\text{)} \\
&= \mathbb{E}\left[\sum_{t=1}^{T}\sum_{i=1}^{n-1}\mathbf{1}\left\{\mathscr{E}_{i,x_t}^{(2)}(t)\right\}\sum_{j=i+1}^{x_t}\Delta_{j-1,j}\right] \qquad (7) \\
&\leq \mathbb{E}\left[\sum_{t=1}^{T}\sum_{i=1}^{n-1}\sum_{j=i+1}^{x_t}\mathbf{1}\left\{\mathscr{E}_{i,j}^{(2)}(t)\right\}\Delta_{j-1,j}\right] \quad \text{(For } i<j_1\leq j_2\text{, } \mathbf{1}\left\{\mathscr{E}_{i,j_1}^{(2)}(t)\right\}\geq\mathbf{1}\left\{\mathscr{E}_{i,j_2}^{(2)}(t)\right\}\text{.)} \\
&\leq \mathbb{E}\left[\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\Delta_{j-1,j}\sum_{t=1}^{T}\mathbf{1}\left\{\mathscr{E}_{i,j}^{(2)}(t)\right\}\right] \\
&= \sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\Delta_{j-1,j}\mathbb{E}[M_{i,j}^{(2)}] \\
&\leq \sum_{1\leq i<j\leq n}\frac{8}{\Delta_{i,j}^{2}}\Delta_{j-1,j}
\end{aligned}
$$

Adding the two bounds gives the statement of the lemma.

Before presenting the next lemma that will finish the proof of Theorem 1, let us make the following definition which will be useful in the proof.

Definition 3 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

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

For an expert i, and $y\geq 0$, let $j_y(i)$ be the maximum numbered expert $j\geq i$ such that $\Delta_{i,j}$ is no more than y. That is

$$j_y(i) := \arg\max\{j : j\geq i, \Delta_{i,j}\leq y\}.$$

Now we are ready to present our next lemma.

Lemma 4 *Let $\Delta_{i,i+1}>0$ for $i=1,2,\ldots,n-1$. Then*

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

Note that this lemma is very important from a technical point of view in the proof of the regret bound for FTAL, but does not have a direct bearing on the intuitive understanding of the algorithm.

Note that Lemma 4 combined with Lemma 2 finishes the proof of Theorem 1. Instead of proving the lemma above, we will prove a slight generalization (that will be useful in taking care of "small $\Delta_{i,i+1}$'s"), and the lemma above will follow as a special case by putting $\varepsilon = 0$.

Let us first motivate the generalization. The left hand side of the first inequality in Lemma 4 can also be written as $\sum_{1\le i<j\le n:\Delta_{i,j}>0} \Delta_{i,j}^{-2}\Delta_{i,i+1}$, since the condition $\Delta_{i,j} > 0$ is vacuous (we are assuming in the statement of the lemma that $\Delta_{i,j} > 0$ for $i < j$). Instead of putting an upper bound on $\sum_{1\le i<j\le n:\Delta_{i,j}>0} \Delta_{i,j}^{-2}\Delta_{i,i+1}$, we will relax the condition $\Delta_{i,j} > 0$ to $\Delta_{i,j} > \varepsilon$ for some $\varepsilon \ge 0$ and prove an upper bound on $\sum_{1\le i<j\le n:\Delta_{i,j}>\varepsilon} \Delta_{i,j}^{-2}\Delta_{i,i+1}$. Let us present the general case.

Lemma 5 *For $\varepsilon \ge 0$,*

$$\sum_{1\le i<j\le n:\Delta_{i,j}>\varepsilon} \Delta_{i,j}^{-2}\Delta_{i,i+1} \le 2\sum_{j=j_0(1)+1}^{n} \max\{\varepsilon, \Delta_{i_0(j)-1,i_0(j)}\}^{-1} \quad \textit{and}$$

$$\sum_{1\le i<j\le n:\Delta_{i,j>\varepsilon}} \Delta_{i,j}^{-2}\Delta_{j-1,j} \le 2\sum_{i=1}^{j_0(n)-1} \max\{\varepsilon, \Delta_{j_0(i),j_0(i)+1}\}^{-1}.$$

Recall from Definition 3 that if $\Delta_{i,j} > 0$ for $i < j$, then $j_0(i) = i$ for all i and $i_0(j) = j$ for all j, and the above lemma reduces to Lemma 4 by taking $\varepsilon = 0$. The more complex bound, in terms of $\Delta_{i_0(j)-1,i_0(j)}$ and $\Delta_{j_0(i),j_0(i)+1}$, will be needed later in the paper when proving the more general Theorem 6 that allows for $\Delta_{i,j} = 0$.

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 $\sum_{j:j>i,\Delta_{i,j}>\varepsilon} \Delta_{i,j}^{-2}$ as follows.

$$\begin{aligned}
\sum_{j:j>i,\Delta_{i,j}>\varepsilon} \Delta_{i,j}^{-2} &= \sum_{j=2}^{n} \mathbf{1}\left\{j > i, \Delta_{i,j} > \varepsilon\right\}\Delta_{i,j}^{-2} \qquad (8)\\
&= \int_{x=0}^{\infty} \left|\left\{j : j > i, \Delta_{i,j} > \varepsilon, \Delta_{i,j}^{-2} \ge x\right\}\right| \mathrm{d}x\\
&= \int_{x=0}^{\infty} \left|\left\{j : \varepsilon < \Delta_{i,j} \le x^{-1/2}\right\}\right| \mathrm{d}x \quad (\Delta_{i,j} > \varepsilon \text{ implies } j > i.)\\
&= -2\int_{y=\infty}^{0} \left|\left\{j : \varepsilon < \Delta_{i,j} \le y\right\}\right| y^{-3}\mathrm{d}y \quad \text{(Changing the variable of integration } x^{-1/2} = y.)\\
&= 2\int_{y=0}^{\infty} \left|\left\{j : \varepsilon < \Delta_{i,j} \le y\right\}\right| y^{-3}\mathrm{d}y. \qquad (9)
\end{aligned}$$

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.)

$$\sum_{i=1}^{n-1} \sum_{j\in\{i+1,i+2,\ldots,n\},\Delta_{i,j}>\varepsilon} \Delta_{i,j}^{-2}\Delta_{i,i+1}$$

$$= \sum_{i=1}^{n-1} \Delta_{i,i+1} \sum_{j: j>i, \Delta_{i,j}>\varepsilon} \Delta_{i,j}^{-2} \tag{10}$$

$$= 2\sum_{i=1}^{n-1} \Delta_{i,i+1} \left(\int_{y=0}^{\infty} \left|\left\{ j : \varepsilon < \Delta_{i,j} \le y \right\}\right| y^{-3} \mathrm{d}y \right) \quad \text{(From (9).)}$$

$$= 2\int_{y=0}^{\infty} y^{-3} \left(\sum_{i=1}^{n-1} \Delta_{i,i+1} \cdot \left|\left\{ j : \varepsilon < \Delta_{i,j} \le y \right\}\right| \right) \mathrm{d}y \quad \text{(Changing the order of integration and summation.)}$$

$$= 2\int_{y=0}^{\infty} y^{-3} \left(\sum_{i=1}^{n-1} \Delta_{i,i+1} \sum_{j=i+1}^{n} \mathbf{1}\left\{ \varepsilon < \Delta_{i,j} \le y \right\} \right) \mathrm{d}y \quad \text{(Expanding } |\{\cdot\}| \text{ into sum of } \mathbf{1}\{\cdot\}.)$$

$$= 2\int_{y=0}^{\infty} y^{-3} \left(\sum_{j=2}^{n} \sum_{i=1}^{j-1} \Delta_{i,i+1} \mathbf{1}\left\{ \varepsilon < \Delta_{i,j} \le y \right\} \right) \mathrm{d}y \quad \text{(Changing the order of summation.)}$$

$$= 2\sum_{j=2}^{n} \int_{y=0}^{\infty} y^{-3} \left(\sum_{i=1}^{j-1} \Delta_{i,i+1} \mathbf{1}\left\{ \varepsilon < \Delta_{i,j} \le y \right\} \right) \mathrm{d}y \quad \text{(Changing the order of summation and integration.)}$$

$$= 2\sum_{j=2}^{n} \int_{y=\varepsilon}^{\infty} y^{-3} \left(\sum_{i=1}^{j-1} \Delta_{i,i+1} \mathbf{1}\left\{ \varepsilon < \Delta_{i,j} \le y \right\} \right) \mathrm{d}y \quad \text{(For } y < \varepsilon \text{, the integrand is 0.)}$$

$$= 2\sum_{j=2}^{n} \int_{y=\varepsilon}^{\infty} y^{-3} \left(\sum_{i=i_y(j)}^{i_\varepsilon(j)-1} \Delta_{i,i+1} \right) \mathrm{d}y \quad \text{(Use Definition 3.)}$$

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

Now, we need a little care in manipulating this expression. Let us consider two cases: (i) $\mu_{i_\varepsilon(j)} = \mu_{i_0(j)}$, which means that there is no arm with mean in $(\mu_j, \mu_j + \varepsilon]$, and (ii) $\mu_{i_\varepsilon(j)} > \mu_{i_0(j)}$, which means that there is some arm with mean in $(\mu_j, \mu_j + \varepsilon]$. In the first case, $\mu_{i_y(j)} - \mu_{i_\varepsilon(j)}$ is zero whenever $y < \Delta_{i_0(j)-1,i_0(j)}$, so the lower limit of the integration can be changed to $\Delta_{i_0(j)-1,i_0(j)}$. In the second case, no special care needs to be taken. Note that in both cases, $\mu_{i_y(j)} - \mu_{i_\varepsilon(j)} \le y$. Also note that for j such that $\mu_j = \mu_1$, the difference $\mu_{i_y(j)} - \mu_{i_\varepsilon(j)}$ is always zero (both terms being equal to μ_1. So, we can change the lower limit of the outer sum to start from $j_0(1)+1$ (the first arm which has mean lower than the mean of the first arm).)

$$\le 2\sum_{j=j_0(1)+1}^{n} \left(\mathbf{1}\left\{ \Delta_{i_0(j)-1,i_0(j)} > \varepsilon \right\} \int_{y=\Delta_{i_0(j)-1,i_0(j)}}^{\infty} y^{-2}\mathrm{d}y + \mathbf{1}\left\{ \Delta_{i_0(j)-1,i_0(j)} \le \varepsilon \right\} \int_{y=\varepsilon}^{\infty} y^{-2}\mathrm{d}y \right)$$

$$= 2\sum_{j=j_0(1)+1}^{n} \left(\mathbf{1}\left\{ \Delta_{i_0(j)-1,i_0(j)} > \varepsilon \right\} \left(\Delta_{i_0(j)-1,i_0(j)} \right)^{-1} + \mathbf{1}\left\{ \Delta_{i_0(j)-1,i_0(j)} \le \varepsilon \right\} (\varepsilon)^{-1} \right)$$

$$= 2\sum_{j=j_0(1)+1}^{n} \left(\max\left\{ \varepsilon, \Delta_{i_0(j)-1,i_0(j)} \right\} \right)^{-1}$$

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 $\varepsilon > 0$ be fixed (the original theorem corresponds to the case $\varepsilon = 0$). Recall the definition of $i_\varepsilon(j)$ and $j_\varepsilon(i)$ from Definition 3. Note that the three conditions: (1) $i < i_\varepsilon(j)$, (2) $j > j_\varepsilon(i)$, and (3) $\Delta_{i,j} > \varepsilon$ 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

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

This can be seen by rewriting Equation (6) as

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

and noting that the second term is at most

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

since only one of the events $\mathscr{E}_{i_t^*,j}^{(1)}(t)$ (corresponding to $j = x_t$) can occur for each t. Equation (7) can be similarly modified by splitting the summation $j = i+1 \ldots x_t$ to $j = i+1 \ldots j_\varepsilon(i)$ and $j = j_\varepsilon(i)+1 \ldots x_t$.

To upper bound the regret by the sum of inverses of $\Delta_{i,i+1}$, we can use Lemma 5. With these modifications to the proof, we have established the following variant of Theorem 1. Note that the result of Theorem 1 can be seen to be a special case of the theorem below by setting $\varepsilon = 0$.

Theorem 6 *For every $\varepsilon \ge 0$, the* FTAL *algorithm has a regret of at most*

$$2\varepsilon T + \sum_{j=j_0(1)+1}^{n} \frac{16}{\max\{\varepsilon, \Delta_{i_0(j)-1,i_0(j)}\}} + \sum_{i=1}^{j_0(n)-1} \frac{16}{\max\{\varepsilon, \Delta_{j_0(i),j_0(i)+1}\}}.$$

with respect to the best ordering.

Remember that the distributions $P_i(\cdot)$ and, in particular, $\varepsilon_0 = \min_i\{\Delta_{j_0(i),j_0(i)+1}\}$ are independent of T. This means that for T large enough ($\mathscr{O}(\varepsilon_0^{-2})$), the optimal ε in the theorem above will be zero, obtaining a constant regret bound with respect to T. Note that to make this statement it is critical to express the regret bound in terms of $\Delta_{j_0(i),j_0(i)+1}$ rather than in terms of $\Delta_{i,i+1}$ to handle the case where $\Delta_{i,i+1} = 0$ for some i, and ensure that ε_0 is bounded away from 0.

2.1.2 Lower Bound

In this section, assuming that the means μ_i are bounded away from 0 and 1, we prove that FTAL's regret 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. Let Bernoulli(p) denote the Bernoulli distribution with mean p. We use $\mathsf{KL}(p;q)$ to denote the KL-divergence of two distributions, and for the case of Bernoulli distributions with means μ and μ', we use the notation $\mathsf{KL}(\mu;\mu')$ instead of writing a somewhat more wordy notation $\mathsf{KL}(\text{Bernoulli}(\mu),\text{Bernoulli}(\mu'))$. Please refer to (Karp & Kleinberg, 2007) and (Cover & Thomas, 1999) for an introduction to KL-divergence.

Lemma 7 *Let $P_i = \text{Bernoulli}(\mu_i)$ for $i = 1,2,\ldots,n$ be the payoff distributions with $\mu_i \in (\alpha,\beta)$ for some $0 < \alpha < \beta < 1$ (μ_i's can be relaxed to lie in the* closed *interval $[\alpha,\beta]$). Let ϕ be any algorithm for the stochastic best expert model. Then, there is an input instance with n arms endowed with some permutation of the aforementioned distributions $(P_1,P_2,\ldots,P_n)$, such that the regret of ϕ up to time T is at least*

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

whenever $T \geq T_0$, where T_0 is a function of n, $(\mu_1,\mu_2,\ldots,\mu_n)$, α, and β.

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

Lemma 8 *Let $P_i = \text{Bernoulli}(\mu_i)$ for $i = 1,2$ be payoff distribution with $\mu_1,\mu_2 \in (\alpha,\beta)$, $\mu_1 > \mu_2$, and $0 < \alpha < \beta < 1$. Let ϕ be an online algorithm for the stochastic best expert problem with two experts. Consider two instances I_1 and I_2 for the stochastic best expert setting: In both instances, there are two experts namely L and R; in I_1, (L,R) are endowed with reward distributions (P_1,P_2) and in I_2, they are endowed with (P_2,P_1). Then the regret of algorithm ϕ on at least one of I_1 or I_2 is*

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

whenever $T \geq T_0$, where $\delta = \mu_1 - \mu_2$, T_0 is a function of (μ_1,μ_2), α, and β, and the constants inside the $\Omega(\cdot)$ may depend on α,β.

Proof Let us define some joint distributions: p is the *joint* distribution in which both experts have payoff distribution P_1, q_L is the distribution in which they have payoff distributions (P_1,P_2) (left is better), and q_R is the distribution in which they have payoff distributions (P_2,P_1) (right expert is better).

Let $T_0 = \frac{c}{\delta^2}$ for $c = \frac{\min\{\alpha(1-\alpha),\beta(1-\beta)\}}{25}$, and $T \geq T_0$. We will prove that if ϕ runs for T rounds, then for one the instances q_L or q_R, it will suffer at least $\Omega(\delta^{-1})$ regret.

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 distribution induced by ϕ on the t-step histories, 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

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

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

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

by the choice of c.

Karp and Kleinberg (2007) prove the following lemma. If there is an event E with $p(E) \geq 1/3$ and $q(E) < 1/3$, then

$$\mathsf{KL}(p;q) \geq \frac{1}{3}\ln\left(\frac{1}{3q(E)}\right) - \frac{1}{e}. \tag{11}$$

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. We next prove the lower bound for n experts.

Proof of Lemma 7: 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 8 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 8, 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. □

2.2 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 the UCB1 algorithm (Auer et al., 2002a), and then present a matching lower bound based on an idea from Lai and Robbins (1985).

2.2.1 Upper Bound (Algorithm: AUER*)*

Here the optimal algorithm is again a natural extension of the UCB1 algorithm (Auer et al., 2002a) 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 (radius) $\rho_{j,t}=\sqrt{\frac{8\ln t}{n_{j,t}}}$ 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 $\hat{\mu}_{j,t}+\rho_{j,t}$ where $\hat{\mu}_{j,t}$ is the mean of the observed rewards of arm j up to time t, and $\rho_{j,t}=\sqrt{\frac{8\ln t}{n_{j,t}}}$ is the width of the confidence interval around arm j at 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 ∈ [n].
for t = 1 to T do
    if ∃ j ∈ A_t s.t. n_j = 0 then
        Play arm x(t) = j
    else
        Play arm x(t) = arg max_{i∈A_t} ( z_i/n_i + sqrt(8 log t / n_i) )
    end
    Observe payoff r_{x(t)}(t) for arm x(t)
    z_{x(t)} ← z_{x(t)} + r_{x(t)}(t)
    n_{x(t)} ← n_{x(t)} + 1
end
```

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

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

Lemma 9 *With the definition of $n_{i,t}$, μ_i, $\hat{\mu}_i$, and $\rho_{i,t} = \sqrt{\frac{8\ln t}{n_{i,t}}}$ the following holds for all $1 \le i \le n$ and $1 \le t \le T$:*

$$\mathbb{P}\Big[\mu_i \in [\hat{\mu}_{i,t} - \rho_{i,t}, \hat{\mu}_{i,t} + \rho_{i,t}]\Big] = \mathbb{P}\Big[\hat{\mu}_{i,t} \in [\mu_i - \rho_{i,t}, \mu_i + \rho_{i,t}]\Big] \ge 1 - \frac{1}{t^4}.$$

Proof The equality follows since the two events are the same. The proof of inequality is an application of Chernoff-Hoeffding bounds, and follows from (Auer et al., 2002a, pp. 242–243).

Theorem 10 *For problem instances with $\Delta_{i,i+1} > 0$ for $i = 1, 2, \ldots, n-1$, the regret of the* AUER *algorithm is at most*

$$(66 \ln T + \mathcal{O}(1)) \cdot \sum_{i=1}^{n-1} \frac{1}{\Delta_{i,i+1}}.$$

up to time T.

The theorem follows immediately from the following lemma and Lemma 4. Note that we are only considering problem instances in which different arms have different means. This restriction will be removed at the end of this section, where we present a general bound, and the above theorem will follows as a special case of the general result.

Lemma 11 *For problem instances with $\Delta_{i,i+1} > 0$ for $i = 1, 2, \ldots, n-1$, the* AUER *algorithm has a regret of at most*

$$(33 \ln T + \mathcal{O}(1)) \cdot \sum_{j=2}^{n} \sum_{i=1}^{j-1} \left(\frac{1}{\Delta_{i,j}^2}\right) \Delta_{i,i+1},$$

up to time T.

Proof We bound the regret of the algorithm arm by arm. Let us consider an arm $2 \le j \le n$. For $i < j$, let us count the number $N_{i,j}$ of times j was played when some arm in $1, 2, \ldots, i$ was awake. (In these iterations, the regret accumulated is at least $\Delta_{i,j}$ and at most $\Delta_{1,j}$.) We claim that $\mathbb{E}[N_{i,j}] \le Q_{i,j}$, where $Q_{i,j} := \frac{33 \ln T}{\Delta_{i,j}^2}$.

We want to claim that after playing j for $Q_{i,j}$ number of times, we are unlikely to make the mistake of choosing j instead of something from the set $\{1, 2, \ldots, i\}$; that is, if the set of awake arms at time t includes some arm in $[i]$ as well as arm j, then with probability at least $1 - \frac{2}{t^4}$, some awake arm in $[i]$ will be chosen rather than arm j.

Let us bound the expected number of times j is chosen when $A_t \cap [i] \ne \emptyset$ and j has already been played $Q_{i,j}$ number of times.

$$\begin{aligned}
&\sum_{Q_{i,j} < s \le t \le T} \mathbb{P}\Big[(x_t = j) \wedge (j \text{ is played } s\text{-th time}) \wedge (A_t \cap [i] \ne \emptyset)\Big] \\
&\le \sum_{Q_{i,j} < s \le t \le T} \mathbb{P}\Big[(x_t = j) \wedge (n_{j,t} = s) \wedge \left(\forall_{k=1}^{i} \left(\hat{\mu}_{j,t} + \rho_{j,t} \ge \hat{\mu}_{k,t} + \rho_{k,t}\right)\right)\Big] \\
&= \sum_{Q_{i,j} < s \le t \le T} \mathbb{P}\left[(x_t = j) \wedge (n_{j,t} = s) \wedge \left(\forall_{k=1}^{i} \left(\hat{\mu}_{j,t} + \sqrt{\frac{8 \ln t}{s}} \ge \hat{\mu}_{k,t} + \rho_{k,t}\right)\right)\right] \\
&\le \sum_{Q_{i,j} < s \le t \le T} \mathbb{P}\left[\forall_{k=1}^{i} \left(\hat{\mu}_{j,t} + \sqrt{\frac{8 \ln t}{s}} \ge \hat{\mu}_{k,t} + \rho_{k,t}\right)\right]. \qquad (12)
\end{aligned}$$

Let us define the event inside the probability expression as E_1 and define E_2 to be the event that $\hat{\mu}_{k,t} \in [\mu_k - \rho_{k,t}, \mu_k + \rho_{k,t}]$ for all $k \in \{j\} \cup \{1,2,\ldots,i\}$. (Although E_1 and E_2 depend on s and t, we suppress this dependence for notational convenience.) The probability of event E_2 is at least $1-(i+1)t^{-4}$ (from Lemma 9).

We will bound use the probability of E_1 by conditioning it on the event E_2. We can write $\mathbb{P}[E_1] = \mathbb{P}[E_1|E_2]\mathbb{P}[E_2] + \mathbb{P}[E_1|E_2^c]\mathbb{P}[E_2^c] \leq \mathbb{P}[E_1|E_2] + \mathbb{P}[E_2^c]$. To bound $\mathbb{P}[E_1|E_2]$, notice that the confidence $\rho_{j,t}$ of arm j is at most $\sqrt{\frac{8\ln T}{33\ln T} \cdot \Delta_{i,j}^2} < \frac{\Delta_{i,j}}{2}$.

If event E_2 happens, $\hat{\mu}_{j,t} + \rho_{j,t} \leq (\mu_j + \rho_{j,t}) + \rho_{j,t} < \mu_j + \Delta_{i,j} = \mu_i$. Also, $\hat{\mu}_{k,t} + \rho_{k,t} \geq \mu_k$ for all $k = 1,2,\ldots,i$. Therefore, the sum in (12) can be upper-bounded by following.

$$
\begin{aligned}
&\sum_{Q_{i,j} < s \leq t \leq T} \left(\mathbb{P}\left[\forall_{k=1}^{i} \left(\hat{\mu}_{j,t} + \sqrt{\frac{8\ln t}{s}} \geq \hat{\mu}_{k,t} + \rho_{k,t} \right) \,\middle|\, E_2 \right] + \mathbb{P}[E_2^c] \right) \\
&\leq \sum_{Q_{i,j} < s \leq t \leq T} \left(\mathbb{P}\left[\forall_{k=1}^{i} (\hat{\mu}_{j,t} + \rho_{j,t} \geq \mu_k) \right] \right) + \sum_{Q_{i,j} < s \leq t \leq T} \frac{i+1}{t^4} \\
&\leq \sum_{Q_{i,j} < s \leq t \leq T} \mathbb{P}\left[\hat{\mu}_{j,t} + \rho_{j,t} \geq \mu_i \right] + \sum_{Q_{i,j} < s \leq t \leq T} \frac{i+1}{t^4} \quad \text{(Since } \mu_1 \geq \mu_2 \geq \cdots \geq \mu_i\text{.)} \\
&\leq \mathscr{O}(nT^{-2}) \quad \text{(The first term is zero, since} \\
&\qquad\qquad \hat{\mu}_{j,t} + \rho_{j,t} < \mu_i\text{, see above.)} \\
&= \mathscr{O}(1).
\end{aligned}
$$

Therefore, after j has been played $Q_{i,j}$ number of times, the expected number of additional times that j is played when $A_t \cap [i] \neq \emptyset$ is bounded above by a constant. This implies

$$
\mathbb{E}[N_{i,j}] \leq Q_{i,j} + \mathscr{O}(1) \leq \frac{33\ln(T)}{\Delta_{i,j}^2} + \mathscr{O}(1).
$$

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

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

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

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

This gives the statement of the lemma.

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

$$
\begin{aligned}
&\sum_{j=2}^{n} \sum_{i=1}^{j-1} (N_{i,j} - N_{i-1,j}) \Delta_{i,j} \\
&= \sum_{j=2}^{n} \sum_{i=1}^{i_\varepsilon(j)} (N_{i,j} - N_{i-1,j}) \Delta_{i,j} \; + \; \sum_{j=2}^{n} \sum_{i=i_\varepsilon(j)+1}^{j-1} (N_{i,j} - N_{i-1,j}) \Delta_{i,j}
\end{aligned}
$$

$$\leq \sum_{j=2}^{n} \sum_{i=1}^{i_\varepsilon(j)-1} N_{i,j}\Delta_{i,i+1} + \sum_{j=2}^{n} N_{i_\varepsilon(j),j}\Delta_{i_\varepsilon(j),j} + \sum_{j=2}^{n} \sum_{i=i_\varepsilon(j)+1}^{j-1} (N_{i,j} - N_{i-1,j})\varepsilon$$
$$\leq \sum_{j=2}^{n} \sum_{i=1}^{i_\varepsilon(j)-1} N_{i,j}\Delta_{i,i+1} + \varepsilon \sum_{j=2}^{n} N_{i_\varepsilon(j),j} + \varepsilon \sum_{j=2}^{n} (N_{j-1,j} - N_{i_\varepsilon(j),j})$$
$$\leq \sum_{1\leq i<j\leq n, \Delta_{i,j}>\varepsilon} N_{i,j}\Delta_{i,i+1} + \varepsilon T,$$

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

Taking the expectation, and using the Lemma 5, we get the following regret bound for AUER algorithm.

Theorem 12 *For any $\varepsilon \geq 0$, the regret of the* AUER *algorithm is at most*

$$\varepsilon T + \sum_{j=j_0(1)+1}^{n} \frac{33\ln T + \mathcal{O}(1)}{\max\{\varepsilon, \Delta_{i_0(j)-1,i_0(j)}\}} + \sum_{i=1}^{j_0(n)-1} \frac{33\ln T + \mathcal{O}(1)}{\max\{\varepsilon, \Delta_{j_0(i),j_0(i)+1}\}},$$

up to time T.

2.2.2 Lower bound

In this section, we prove that the AUER algorithm presented is information theoretically optimal up to constant factors when the numbers μ_i — the mean payoffs of arms — 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 (1985) for two-armed bandit problems. The difference is that Lai and Robbins prove their lower bound only in the case when $T \to \infty$, but we want to get bounds that hold for finite T. Our main result is stated in the following lemma.

Lemma 13 *Let $P_i = \text{Bernoulli}(\mu_i)$ for $i = 1,2,\ldots,n$ be payoff distributions with $\mu_i \in (\alpha,\beta)$ for some $0 < \alpha < \beta < 1$. Let ϕ be an algorithm for picking among n arms such that for all t, the expected number of times ϕ plays a suboptimal bandit up to time t is bounded above by $c_1 t^{0.1} + c_2$ (c_1 and c_2 possibly depend on μ_i). Then, there is an input instance with n arms endowed with some permutation of the aforementioned distributions $(P_i)_{i=1}^n$, such that the regret of ϕ is 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 $T \geq T_0$, where T_0 is a function of n, μ_i, c_1, c_2, α, β.

We note that the exponent 0.1 in the lemma is quite arbitrary. Indeed, any nonzero exponent would work for the purpose of the proof.

Note that the above lower bound without the $(\log T)$ factor follows from the stochastic best expert lower bound in Lemma 7.

Using the fact that for $\mu_i \in (\alpha,\beta)$, $\mathsf{KL}(\mu_j;\mu_i) = \mathscr{O}_{\alpha,\beta}(\Delta_{i,j}^2)$, the lower bound can also be stated as

$$\Omega_{\alpha,\beta}\left(\sum_{i=1}^{n-1}\frac{(\log T)}{\Delta_{i,i+1}}\right),$$

which matches (up the constant factors) the upper bound in Theorem 10. Note that the notations $\mathscr{O}_{\alpha,\beta}(\cdot)$ and $\Omega_{\alpha,\beta}(\cdot)$ hide dependence on α and β.

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 14 *Let $P_i = \text{Bernoulli}(\mu_i)$ for $i = 1,2$ with $\mu_2 < \mu_1$, $\mu_i \in (\alpha,\beta)$ for $i = 1,2$ and $0 < \alpha < \beta < 1$. Let ϕ be any algorithm for choosing among two arms which never picks the worse arm (for any values of μ_1 and μ_2 in (α,β)) more than $c_1 t^{0.1} + c_2$ times up to time t (c_1 and c_2 possibly depend on μ_1 and μ_2). Then there exists an instance with two arms endowed with two distributions above (in some order) such that the regret of the algorithm ϕ when presented with this instance is at least*

$$\frac{1}{6}\left(\frac{(\log T)(\mu_1-\mu_2)}{\mathsf{KL}(\mu_2;\mu_1)}\right),$$

for all $T \geq T_0$, and the value of T_0 can be explicitly computed as a function of μ_1, μ_2, c_1, c_2, α, β.

Proof 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

$$|\mathsf{KL}(\mu_2;\lambda) - \mathsf{KL}(\mu_2;\mu_1)| \leq \frac{1}{10}\cdot\mathsf{KL}(\mu_2;\mu_1),$$

because of the continuity of KL divergence in its second argument. Indeed, using the convexity of $\mathsf{KL}(\mu_2;\cdot)$ (for fixed μ_2), and the fact that the slope of $\mathsf{KL}(\mu_2;\cdot)$ is bounded by $\frac{\beta-\mu_2}{\beta(1-\beta)}$, λ can be chosen to be $\min\left\{\mu_1 + \frac{\mathsf{KL}(\mu_2;\mu_1)}{10}\frac{\beta(1-\beta)}{(\beta-\mu_2)}, \frac{\beta-\mu_1}{2}\right\}$. This claim provides us with a Bernoulli distribution with mean λ (which is an explicit function of μ_i and β) such that

$$\mathsf{KL}(\mu_2;\lambda) \leq \frac{11}{10}\cdot\mathsf{KL}(\mu_2;\mu_1). \tag{14}$$

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 the algorithm ϕ playing against Bernoulli arms with means (μ_1,μ_2), and q is the distribution induced by ϕ playing against Bernoulli arms with means (μ_1,λ). From the assumption of the lemma, we have

$$\mathbb{E}_q[T - n_{2,T}] \leq c_1 T^{0.1} + c_2.$$

Note that c_1 and c_2 here are functions of μ_1 and λ (which in turn is a function of μ_i, α, β). By an application of Markov's inequality, we get that

$$\mathbb{P}_q\left[n_{2,T} < \frac{9}{10}(\log T)/\mathsf{KL}(\mu_2;\lambda)\right] \leq \frac{\mathbb{E}_q[T-n_{2,T}]}{T - \frac{9}{10}(\log T)/\mathsf{KL}(\mu_2;\lambda)}$$

$$\begin{aligned} &\leq \frac{c_1 T^{0.1} + c_2}{T/2} \quad \left(\text{for } T > e^{5/(9\mathsf{KL}(\mu_2;\lambda))}\right) \\ &\leq 4c_1 T^{-0.9}. \qquad (\text{for } T > (c_2/c_1)^{10}) \end{aligned} \tag{15}$$

Let $\mathscr{E}$ denote the event that $n_{2,T} < \frac{9}{10}(\log T)/\mathsf{KL}(\mu_2;\lambda)$. If $\mathbb{P}_p(\mathscr{E}) < 1/3$, then

$$\begin{aligned} \mathbb{E}_p[n_{2,T}] &\geq \mathbb{P}_p(\overline{\mathscr{E}}) \cdot \frac{9}{10}\left(\log T\right)/\mathsf{KL}(\mu_2,\lambda) \\ &\geq \frac{2}{3} \cdot \frac{9}{10} \cdot \frac{\log T}{\mathsf{KL}(\mu_2,\lambda)} \\ &\geq \frac{2}{3} \cdot \frac{9}{11} \cdot \frac{\log T}{\mathsf{KL}(\mu_2;\mu_1)}, \end{aligned}$$

which implies the stated lower bound.

Henceforth, we will assume $\mathbb{P}_p(\mathscr{E}) \geq 1/3$. We have $\mathbb{P}_q(\mathscr{E}) < 1/3$ using (15). Now applying the lemma from (Karp & Kleinberg, 2007) stated in (11), we have

$$\begin{aligned} \mathsf{KL}(p;q) &\geq \frac{1}{3}\ln\left(\frac{1}{3 \cdot 4c_1 T^{-0.9}}\right) - \frac{1}{e} \\ &= \frac{1}{3}(0.9)\ln T - \left(\frac{1}{e} + \frac{1}{3}\ln(12c_1)\right) \\ &= (0.3)\ln T - \left(\frac{1}{3}\ln(e^{3/e}c_1)\right). \end{aligned} \tag{16}$$

The chain rule for KL divergence (Cover & Thomas, 1999, Theorem 2.5.3) implies

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

Combining (16) with (17), we get

$$\begin{aligned} \mathbb{E}_p[n_{2,T}] &\geq \frac{(0.3)\ln T - \frac{1}{3}\ln(e^{3/e}c_1)}{\mathsf{KL}(\mu_2;\lambda)} \\ &\geq \frac{0.3}{1.1}\frac{\ln T}{\mathsf{KL}(\mu_2;\mu_1)} - \frac{1}{3}\frac{\ln(e^{3/e}c_1)}{\mathsf{KL}(\mu_2;\mu_1)} \\ &= \frac{3}{11}\frac{\ln(T)}{\mathsf{KL}(\mu_2;\mu_1)} - \frac{1}{3}\frac{\ln(e^{3/e}c_1)}{\mathsf{KL}(\mu_2;\mu_1)} \\ &\geq \frac{3}{11}\frac{\ln(T)}{\mathsf{KL}(\mu_2;\mu_1)} - \frac{1}{10}\frac{\ln(T)}{\mathsf{KL}(\mu_2;\mu_1)} \quad (\text{for } T > (e^{3/e}c_1)^{10/3}) \\ &\geq \frac{1}{6}\frac{\ln(T)}{\mathsf{KL}(\mu_2;\mu_1)}. \end{aligned}$$

This gives the required regret bound. The explicit value of T above which the bound holds is

$$T_0 := \max\left\{e^{5/(9\mathsf{KL}(\mu_2;\lambda))}, \left(\frac{c_2}{c_1}\right)^{10}, (e^{3/e}c_1)^{10/3}\right\},$$

which can be explicitly written as a function of μ_1, μ_2, c_1, c_2, α, β.

We now extend the result from 2 to n bandits.

Proof of Lemma 13: 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 bounds as done in the proof of Lemma 7.

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

$$\Omega\left(\log\left(\frac{T}{n}\right)\sum_{i=1}^{\lfloor n/2\rfloor}\left(\frac{\mu_{2i-1}-\mu_{2i}}{\mathsf{KL}(\mu_{2i};\mu_{2i-1})}\right)\right).$$

We get a similar lower bound by presenting the algorithm with $(2i,2i+1)$:

$$\Omega\left(\log\left(\frac{T}{n}\right)\sum_{i=1}^{\lfloor (n-1)/2\rfloor}\left(\frac{\mu_{2i}-\mu_{2i+1}}{\mathsf{KL}(\mu_{2i+1};\mu_{2i})}\right)\right).$$

Taking the average of the two lower bounds and $T \geq n^2$ gives the required lower bound of

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

finishing the proof of the lemma. □

3 Adversarial Model of Rewards

We now turn our attention to the case where no distributional assumptions are made on the generation of rewards. We consider in turn the best expert setting and the multi-armed bandit setting. For each setting, we first prove information theoretic lower bounds on the regret of any online learning algorithm, and then present online algorithms whose regret is within a constant factor of the lower bound for the expert setting and within a sub-logarithmic 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 $\mathcal{O}(T^{1-\varepsilon}n^c)$ for some positive constants ε, c.

3.1 Best Expert Setting

In this section, we consider the adversarial sleeping best expert setting. Recall that in the sleeping best expert setting, the algorithm chooses an expert to play in each time round from the set of available experts, and at the end of the round, gets to observe the rewards of *all* available experts for that round, not just for the one it chose. There is no assumption on how the rewards of these experts are generated in each round; indeed an adversary chooses the reward of each expert in each time round, and can observe the choices made by the algorithm prior to that round in choosing the rewards for a particular round. Additionally, the adversary also chooses which subset of the experts will be awake (available) in each time round.

We first present a lower bound on the achievable regret of any algorithm for the adversarial sleeping best expert problem.

Theorem 15 *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 of length T) 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 (19). 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 total payoffs of x_i and y_i, respectively, during the two-experts game, it follows from Khintchine's inequality (Khintchine, 1923) that

$$\mathbb{E}\left(\left|g^{(i)}(x_i)-g^{(i)}(y_i)\right|\right)=\Omega\left(\sqrt{T_0}\right). \tag{18}$$

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

$$Q(n)=2Q(n/2)+n/2, \tag{19}$$

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 (18), 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

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

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, i.e. it makes these choices without considering the algorithm's past decisions. It only needs to be able to carefully coordinate which experts are awake based on the payoffs at previous time steps.

Even more interesting is the fact that 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 $\mathcal{O}(\sqrt{Tn \log(n)})$. We turn our attention to this algorithm now.

To achieve this regret we transform the sleeping experts problem to a problem with $n!$ experts that are always awake, and we choose among these $n!$ experts using the Hedge algorithm (Freund & Schapire, 1999). In the transformed problem, we have one expert for each σ-policy (i.e. ordering of the original n experts). At each round, each of the $n!$ experts makes a prediction according to its corresponding σ-policy, (i.e. the same prediction as the highest ranked awake expert in the corresponding ordering), and receives the payoff of that policy (i.e. the payoff of the highest ranked awake expert in the corresponding ordering).

Theorem 16 *An algorithm that makes predictions using the* Hedge *algorithm on the transformed problem achieves $\mathcal{O}(\sqrt{Tn \log(n)})$ 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 $\mathcal{O}(\sqrt{T \log(n!)})$ with respect to the best expert in the transformed problem, the same regret is achieved by the algorithm in the original problem. The theorem follows by applying the bound $\log(n!) = \mathcal{O}(n \log n)$, which is a consequence of Stirling's formula.

In a naive implementation the algorithm described above is obviously not computationally efficient since in each round we have to sample among $n!$ experts and update $n!$ weights. A natural question is whether this algorithm can be implemented in polynomial time by devising an efficient sampling scheme and a clever weight update procedure. The following theorem, unfortunately, shatters any hope that this might be possible.

Theorem 17 *Unless* RP = NP, *any learning algorithm for the adversarial sleeping experts problem that:*

1. *generates its output by sampling over σ-policies, independently of the set of awake experts*
2. *has regret bounded by $T^{1-\varepsilon} \cdot p(n)$ for some $\varepsilon > 0$ and some polynomial function $p(\cdot)$*

cannot be implemented in polynomial time.

Proof We prove this theorem via a reduction from the minimum feedback arc set problem (Garey & Johnson, 1979). The notion of reduction here is not the usual Karp-reduction, but we will show that if there is an algorithm with specified conditions, then we can find the optimum for any feedback arc set instance with probability at least $1-\delta$ for any constant $\delta > 0$.

Let ALG be any algorithm that respects the conditions in the theorem. We are given a directed graph $G = (V,A)$, in which we are to find the minimum feedback arc set. Every permutation of the vertices defines a feedback arc set, but this mapping is not one to one. (There can be many permutations for one feedback arc set.) For a permutation σ, the corresponding feedback arc set is the set of arcs going from higher numbered vertices to lower numbered vertices, i.e., $\{a = (u,v) \in A : \sigma(u) > \sigma(v)\}$. The cardinality of this set is denoted by $\texttt{FAS}(\sigma)$. For a feedback arc set $A' \subseteq A$, a corresponding permutation can be found by choosing one of the topological orderings of the graph $(G, A \setminus A')$. It is easy to see that the minimum feedback arc set is equal to $\min_\sigma \texttt{FAS}(\sigma)$. We will use the learning algorithm ALG to find, with high probability, an ordering σ minimizing $\texttt{FAS}(\sigma)$.

We instantiate an adversarial sleeping experts problem with $|V|$ experts, one for each vertex in the graph. In each round, the adversary selects an arc (u,v) in A uniformly at random and makes the two experts corresponding to the head (v) and the tail (u) of the selected arc awake and all the other experts asleep. It then associates a payoff of 1 to the expert corresponding to the tail of the arc and a payoff of 0 to the expert corresponding to the head of the arc. We play for $T := 2(\lceil \frac{1}{\delta} \rceil p(n)m)^{1/\varepsilon}$ rounds and in each round we record the σ-policy selected by ALG and also the feedback arc set value of the permutation σ. At the end of the T rounds we choose the best permutation among the T rounds — the one with the smallest $\texttt{FAS}(\sigma)$ value — and output the corresponding feedback arc set. See Algorithm 3.

```
1  Let σ = (1,2,...,|V|) (current best permutation) and x = FAS(σ) (value of the best feedback arc set
   so far).
2  for t = 1 to T = 2(⌈1/δ⌉ p(n)m)^{1/ε} do
3      Choose (u,v) ∈ A at random from m arcs in A. Let {u,v} be the set of awake experts. Set the
       payoff of u to 1 and the payoff of v to 0.
4      Record the permutation σ_t that the algorithm ALG outputs.
5      if x > FAS(σ_t) then
6          σ ← σ_t
7          x ← FAS(σ_t)
8      end
9  end
10 Output {a = (u,v) ∈ A : σ(u) > σ(v)} as the feedback arc set.
```

Algorithm 3: Algorithm to solve Feedback Arc Set Problem from low regret adversarial expert algorithm.

Let $\texttt{FAS}_*$ be the optimum value of the feedback arc set. Let σ be the permutation selected by Algorithm 3. We claim that $\texttt{FAS}(\sigma) = \texttt{FAS}_*$ with probability at least $1-\delta$. Since the number of rounds is polynomial in n and m (ε and δ are constants), this will solve feedback arc set in randomized polynomial time.

Since the expected regret of the algorithm is at most $T^{1-\varepsilon}p(n)$, it follows from Markov's inequality that with probability at least $1-\delta$, the regret is at most $\frac{1}{\delta}T^{1-\varepsilon}p(n)$. We will prove that in this event, our algorithm finds a σ with $\texttt{FAS}(\sigma) = \texttt{FAS}_*$.

We prove this claim by contradiction. If not, then for all $t = 1,2,\ldots,T$, $\texttt{FAS}(\sigma_t) \geq \texttt{FAS}_* + 1$. The expected reward of choosing a permutation τ is $1 - \frac{\texttt{FAS}(\tau)}{m}$. Therefore the expected

regret in each round is at least

$$\left(1 - \frac{\texttt{FAS}_*}{m}\right) - \left(1 - \frac{\texttt{FAS}(\sigma_t)}{m}\right) \geq \frac{1}{m}.$$

Hence, the total regret of the algorithm is at least $T \cdot \frac{1}{m}$. We also know that the regret is at most $\frac{1}{\delta} T^{1-\varepsilon} p(n)$. This gives the following relation:

$$\frac{T}{m} \leq \frac{1}{\delta} T^{1-\varepsilon} p(n),$$

which simplifies to $T \leq (\frac{1}{\delta} p(n) m)^{1/\varepsilon}$, a contradiction since we have taken T to be twice as much. This proves that if we run our algorithm for $T := 2(\lceil \frac{1}{\delta} \rceil p(n) m)^{1/\varepsilon}$, then with probability at least $1 - \delta$, we recover the optimum feedback arc set for the graph. This proves the theorem.

Note that this does not mean that there does not exist an efficient, low regret algorithm for the adversarial sleeping experts problem. One might be able to design an efficient, low regret algorithm that either does not sample over σ-policies, or makes the sampling dependent on the set of awake experts. For instance, there exists a simple algorithm that achieves low regret against the particular adversary used in the proof above: run a separate instance of the Hedge algorithm for every pair of experts and, in each round, use the instance of Hedge corresponding to the two experts that are awake. Since the adversary will only present the algorithm with two awake experts at a time, this algorithm can always make a prediction, and its regret will be bounded by $\mathcal{O}(\sqrt{T \cdot n^2})$.

3.2 Multi-Armed Bandit Setting

Finally, we consider the adversarial sleeping multi-armed bandit setting. Recall that in the sleeping multi-armed bandit setting, the algorithm chooses an arm to play in each round from the set of available arms, and at the end of the round, gets to observe the rewards of the *chosen* arm (unlike best expert setting, where the algorithm observes the reward of all potential choices). There is no assumption on how the rewards of these arms are generated in each round. Additionally, an adversary also chooses which subset of arms will be awake (available to be chosen by the algorithm) in each round.

We first present a lower bound on the achievable regret of any algorithm for the adversarial sleeping multi-armed bandit problem.

Theorem 18 *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 standard multi-armed bandit when all the bandits are awake (Auer et al., 2002b). In the standard "all-awake" 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 bandit. 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 $\frac{1}{2} + \varepsilon$ 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. Consequently, the regret of any learning algorithm with respect to the best ordering is bounded below by a positive constant times the following expression:

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

The theorem follows.

Let us now turn our attention to getting an algorithm for the adversarial sleeping multi-armed bandit problem. To get an upper bound on regret, we will use the Exp4 algorithm (Auer et al., 2002b). Since Exp4 only works against oblivious adversaries, in what follows we will also assume an oblivious adversary (i.e. the rewards for each arm at each round do not depend on the past choices of the algorithm).

Exp4 chooses an arm by combining the advice of a set of "experts". At each round, each expert provides advice in the form of a probability distribution over arms. 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 arms.)

To use Exp4 for the sleeping experts setting, we concoct $n!+1$ "experts", one corresponding to the "uniform" expert which chooses each arm with equal probability, and one each for $n!$ orderings. The expert corresponding to an ordering σ always "advises" to play the arm $\sigma(A_t)$ (first available arm in its ordering), i.e., in each round, the advice of expert σ is a point distribution concentrated on the highest ranked arm in the corresponding ordering σ.

This introduces a slight problem. Since the uniform expert may advise us to pick arms which are not awake, we assume for convenience that the algorithms is *not* restricted to choose an action from A_t (awake set), but 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 (and don't use the feedback obtained about its payoff). Such a transformation can only increase the algorithm's payoff, i.e. decrease the regret. Hence, to prove the regret bound asserted in Theorem 19 below, it suffices to prove the same regret bound for the case when algorithm is allowed to choose an arm in complement of A_t with zero payoff.

Theorem 19 *The* Exp4 *algorithm as described above achieves a regret of* $\mathcal{O}(n\sqrt{T\log(n)})$ *with respect to the best ordering, against an oblivious adversary.*

Proof We have n arms and $1+n!$ experts, so the regret of Exp4 with respect to the payoff of the best expert is $\mathcal{O}(\sqrt{Tn\log(n!+1)})$ (Auer et al., 2002b). Using the estimate $\log(n!+1) = \mathcal{O}(n\log n)$, this regret bound can be rewritten as $\mathcal{O}(n\sqrt{T\log n})$. 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 gap resulting from adapting the Exp4 algorithm to our setting. In the classical multi-armed bandit setting, Audibert and Bubeck (2009) closed a similar gap ($\mathcal{O}(\sqrt{Tn\log n})$ upper bound versus $\Omega(\sqrt{Tn})$ lower bound) by improving the Exp3 algorithm. It is not clear how the policies from (Audibert & Bubeck, 2009) can be adapted for Exp4 algorithm, so closing the $\mathcal{O}(\sqrt{\log n})$ gap in the sleeping multi-armed bandit problem setting remains an important open problem.

4 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.

Acknowledgements

We are very grateful to anonymous reviewers for COLT 2008 and the COLT'08 Special Issue for the Machine Learning Journal for providing us with thorough and thoughtful comments. They were great help in improving the presentation and readability of the paper.

References

Audibert, J.-Y., & Bubeck, S. (2009). Minimax policies for adversarial and stochastic bandits. In *Colt*.

Auer, P., Cesa-Bianchi, N., & Fischer, P. (2002a). Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, *47*, 235–256.

Auer, P., Cesa-Bianchi, N., Freund, Y., & Schapire, R. E. (2002b). The nonstochastic multiarmed bandit problem. *SIAM J. Comput.*, *32*, 48–77.

Azuma, K. (1967). Weighted sums of certain dependent random variables. *Tohoku Math. J.*, *19*, 357–367.

Blum, A., & Mansour, Y. (2005). From external to internal regret. *COLT* (pp. 621–636).

Cesa-Bianchi, N., Freund, Y., Haussler, D., Helmbold, D. P., Schapire, R. E., & Warmuth, M. K. (1997). How to use expert advice. *J. ACM*, *44*, 427–485.

Cover, T. M., & Thomas, J. A. (1999). *Elements of information theory*. J. Wiley.

Freund, Y., & Schapire, R. E. (1999). Adaptive game playing using multiplicative weights. *Games and Economic Behavior*, *29*, 79–103.

Freund, Y., Schapire, R. E., Singer, Y., & Warmuth, M. K. (1997). Using and combining predictors that specialize. *STOC* (pp. 334–343).

Garey, M. R., & Johnson, D. S. (1979). *Computers and intractability: A guide to the theory of NP-completeness*. W. H. Freeman and Company.

Gittins, J. C. (1979). Bandit processes and dynamic allocation indices. *Journal of the Royal Statistical Society, Series B, 41*, 148–177.

Gittins, J. C., & Jones, D. M. (1979). A dynamic allocation index for the discounted multi-armed bandit problem. *Biometrika, 66*, 561–565.

Hannan, J. (1957). Approximation to Bayes risk in repeated plays. *M. Dresher, A. Tucker, P. Wolfe (Eds.), Contributions to the Theory of Games, Princeton University Press* (pp. 97–139).

Hoeffding, W. (1963). Probability inequalities for sums of bounded random variables. *J. American Stat. Assoc., 58*, 13–30.

Kalai, A. T., & Vempala, S. (2005). Efficient algorithms for on-line optimization. *J. Computer and System Sciences, 71*, 291–307.

Karp, R. M., & Kleinberg, R. (2007). Noisy binary search and its applications. *SODA* (pp. 881–890).

Khintchine, A. (1923). Über dyadische Brüche. *Math Z., 18*, 109–116.

Lai, T. L., & Robbins, H. (1985). Asymptotically efficient adaptive allocations rules. *Adv. in Appl. Math., 6*, 4–22.

Langford, J., & Zhang, T. (2007). The epoch-greedy algorithm for multiarmed bandits with side information. *NIPS*.

Littlestone, N., & Warmuth, M. K. (1994). The weighted majority algorithm. *Inf. Comput., 108*, 212–261. An extended abstract appeared in IEEE Symposium on Foundations of Computer Science, 1989, pp. 256–261.

Robbins, H. (1952). Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society, 58*, 527–535.

Vovk, V. G. (1990). Aggregating strategies. *COLT* (pp. 371–386).

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