Theoretical Machine Learning
Lecture 8

Instructor: Haipeng Luo

1 Multi-armed Bandits

We have discussed how to design algorithms for several online learning problems. All these problems fall into the category of learning with "full information", in the sense that at the end of each round, the decision of the environment z_t is completely revealed to the learner. With this information, the learner can reason about how much loss he/she would have suffered if a different action $\widehat{y}$ would have been taken, by simply evaluating $\ell(\widehat{y}, z_t)$.

While this full-information model captures many problems, there is also a wide range of problems where z_t is not completely revealed and thus the learner is required to learn under partial information or limited feedback. In the rest of this course, we focus on several classic problems in such more challenging settings. Unfortunately, it is not clear how to apply similar minimax machinery we used before to directly study the learnability with partial information. Instead, we will discuss several key algorithmic ideas to tackle these problems.

We start with one of the most fundamental problems in this area: Multi-armed Bandits (MAB), which we briefly mentioned in Lecture 1. Using previous notation, MAB is a problem with $\mathcal{D} = \mathcal{F} = \{1, \ldots, K\} \triangleq [K]$ for some K, $\mathcal{Z} = [0,1]^K$, and $\ell(\widehat{y}, z) = z(\widehat{y})$. Instead of seeing z_t at the end of round t, the learner only observes one coordinate of z_t: $z_t(\widehat{y}_t)$, that is, the coordinate chosen by the learner. For simplicity we only consider oblivious environments, and for conversion we will deploy slightly different notation for the rest of this lecture and describe the problem equivalently as follows: The environment first decides n loss vectors: $\ell_1, \ldots, \ell_n \in [0,1]^K$ (knowing the learner's algorithm). Then for each $t = 1, \ldots, n$, learner selects $a_t \in [K]$, suffers and observes loss $\ell_t(a_t)$.

The name "multi-armed bandits" comes from the original motivation of this problem: imagine a gambler in a casino who has money to play slot machines for n times. The question is then how the gambler should sequentially allocate these n plays to the K available slot machines, with the goal of winning as much as possible. In the formulation above, the vector ℓ_t naturally encodes the loss (equivalently negative reward) of playing each machine at time t, and a_t corresponds to the actual machine that the gambler selects. Of course, after playing this machine, the gambler only observes the loss of this machine, which is $\ell_t(a_t)$, and has no information on what he/she would have received if a different machine was chosen, that is, $\ell_t(a)$ for any other $a \neq a_t$. This aspect is exactly captured by this partial information model.

A slot machine is sometimes called a "one-armed bandit", hence the name multi-armed bandit for this problem. Because of this, we sometimes call each action $a \in [K]$ an "arm". This simple model and its variant in fact capture many real-life applications, with recommendation systems as one of the most notable examples — arms correspond to items to recommend and losses correspond to the user's response (e.g. whether the user clicks on the recommended item or not).

In theory, MAB is also a classic example to understand the trade-off between *exploration* and *exploitation*, which is the key difficulty of this problem. Indeed, on the one hand, it is tempting to select arms that have suffered small losses before (exploitation), but on the other hand, there is also an incentive to select other actions just to find out if they can actually lead to even smaller losses (exploration). Having a good balance between these two is the key to design good algorithms for MAB and in general any other learning problems with partial information.

Before we discuss how to do so, recall the goal of the learner is as usual to minimize regret, defined as

$$\mathrm{Reg}_n = \sum_{t=1}^{n} \ell_t(a_t) - \min_{a\in[K]} \sum_{t=1}^{n} \ell_t(a).$$

So far we have made no assumption on how the loss vectors $\ell_1, \ldots, \ell_n$ are generated. This is called the adversarial setting. However, the *stochastic setting* is equally important in the literature. In a stochastic setting, each arm a has a underlying loss distribution with mean $\mu(a)$, and the losses $\ell_1(a), \ldots, \ell_n(a)$ are i.i.d. samples of this distribution. In this case we usually care about the so-called *pseudo regret*:

$$\overline{\mathrm{Reg}}_n = \mathbb{E}\left[\sum_{t=1}^{n} \mu(a_t) - \min_{a\in[K]} \sum_{t=1}^{n} \mu(a)\right].$$

Compared to $\mathbb{E}\left[\mathrm{Reg}_n\right]$, the difference is that we push the expectation inside the "min", which also means $\overline{\mathrm{Reg}}_n \leq \mathbb{E}\left[\mathrm{Reg}_n\right]$. Dealing with pseudo regret allows us the ignore the deviation of the samples $\ell_1(a), \ldots, \ell_n(a)$ from the mean $\mu(a)$ in the objective, which is natural in the stochastic setting and also allows us the derive tighter bounds as we will see soon. We discuss these two settings in the following two sections respectively.

2 Adversarial MAB

Note that if we could observe the entire loss vector ℓ_t at the end of each round, then this is simply a problem of learning with a finite class, and we have discussed that the Hedge algorithm achieves $\mathcal{O}(\sqrt{n \ln K})$ regret in this case. The difficulty is of course that we do not have the entire loss vector ℓ_t. However, a key technique for dealing with adversarial problems with partial information is to construct some *estimator* of the unknown information, and then plug this into an algorithm for the full information setting. For MAB, this means that we need to construct a good estimator $\widehat{\ell}_t$ for ℓ_t, and then simply plug this into the Hedge algorithm.

How do we construct such estimators? It turns out that as long as the algorithm is randomized, there is a standard estimator. Specifically, if the probability of selecting arm a at time t is $p_t(a)$, then the *importance-weighted estimator* is defined as

$$\forall a \in [K], \quad \widehat{\ell}_t(a) = \frac{\ell_t(a)}{p_t(a)} \mathbf{1}\{a = a_t\} = \begin{cases} \frac{\ell_t(a_t)}{p_t(a_t)} & \text{if } a = a_t, \\ 0 & \text{else.} \end{cases}$$

Clearly the estimator is computable using only the information $\ell_t(a_t)$. More importantly, it is *unbiased*: for any $a \in [K]$,

$$\mathbb{E}_t\left[\widehat{\ell}_t(a)\right] = (1 - p_t(a)) \times 0 + p_t(a)\frac{\ell_t(a)}{p_t(a)} = \ell_t(a)$$

where $\mathbb{E}_t[\cdot]$ is the conditional expectation with respect to the random draw of a_t given everything before round t. We now plug this estimator into the Hedge algorithm and obtain the classic adversarial MAB algorithm called Exp3 (which stands for *Exponential-weight for Exploration and Exploitation*): at time t, sample $a_t \sim p_t \in \Delta(K)$ where

$$\forall a \in [K], \quad p_t(a) \propto \exp\left(-\eta \sum_{\tau=1}^{t-1} \widehat{\ell}_\tau(a)\right) \tag{Exp3}$$

for some learning rate $\eta > 0$.

Before analyzing the regret of this algorithm, let's first see why this algorithm makes sense, and in particular, where is the aforementioned exploration-exploitation trade-off? The exploitation part is basically executed by the Hedge algorithm: arms with smaller estimated losses are selected with higher probability. On the other hand, the exploration part is somewhat implicit. Indeed, whenever an arm a_t is selected (maybe due to exploitation), the probability of selecting this arm next time is always decreased (or at least not increased), which will then encourage the algorithm to explore other actions. This is due to the structure of the estimator $\widehat{\ell}_t$ so that only the selected action a_t can have non-zero loss, while all the other actions have estimated loss 0.

To better understand the importance of this implicit exploration, consider the case where the losses are negative: $\ell_t \in [-1, 0]^K$ (or equivalently their magnitude corresponds to reward). Then Exp3 should not work anymore, since whenever an arm a_t is selected, its probability of being selected next time gets even larger (again due to the structure of $\widehat{\ell}_t$). This clearly lacks sufficient exploration and will suffer linear regret in the worst case.[1]

Now we analyze the regret of Exp3. It might be tempting to conclude that, just like Hedge, Exp3 achieves regret $\mathbb{E}[\text{Reg}_n] = \mathcal{O}(\sqrt{n \ln K})$ as well — after all, we only care about expected regret here and the estimators are all unbiased. However, a closer look at the Hedge analysis (i.e. Lemma 1 of Lecture 6) reveals that its regret is $\mathcal{O}(C\sqrt{n \ln K})$ for losses bounded by $C > 0$. Without loss of generality we have assumed that the true losses are in $[0, 1]$, but how large can $\widehat{\ell}_t(a)$ be? It can in fact be unbounded due to the inverse probability weighting! If we try to explicitly enforce a lower bound γ for the probabilities to make sure that the estimators are never larger than $1/\gamma$ (there are many different ways to do so), then accordingly we will pay extra γT regret due to this constraint. Even trading off γ optimally will at best lead to regret of order $\mathcal{O}(T^{2/3})$ (details omitted).

However, the magic of Hedge is that it somehow only cares about the variance of the estimators (which can still be unbounded as shown below), and more importantly it has some intrinsic variance cancellation effect which eventually allows it to still ensure $\mathcal{O}(\sqrt{T})$ regret. This is shown in the following theorem.

Theorem 1. *With $\eta = \sqrt{\frac{\ln K}{nK}}$, Exp3 ensures $\mathbb{E}\left[\text{Reg}_n\right] = \mathcal{O}(\sqrt{nK \ln K})$.*

Proof. Since $\widehat{\ell}_t(a) \geq 0$ for all t and a, we can directly apply Lemma 1 of Lecture 6 and get for $a^\star \in \text{argmin}_{a \in [K]} \sum_{t=1}^n \ell_t(a)$,

$$\sum_{t=1}^n \left\langle p_t, \widehat{\ell}_t \right\rangle - \sum_{t=1}^n \widehat{\ell}_t(a^\star) \leq \frac{\ln K}{\eta} + \eta \sum_{t=1}^n \sum_{a=1}^K p_t(a) \widehat{\ell}_t^2(a).$$

Noting that the conditional variance (or rather the second moment) of the estimator is $\mathbb{E}_t[\widehat{\ell}_t^2(a)] = p_t(a) \times \frac{\ell_t^2(a)}{p_t^2(a)} = \frac{\ell_t^2(a)}{p_t(a)}$ and taking expectation we have

$$\mathbb{E}\left[\sum_{t=1}^n \ell_t(a_t) - \sum_{t=1}^n \ell_t(a^\star)\right] \leq \frac{\ln K}{\eta} + \eta \mathbb{E}\left[\sum_{t=1}^n \sum_{a=1}^K p_t(a) \frac{\ell_t^2(a)}{p_t(a)}\right] \leq \frac{\ln K}{\eta} + \eta n K.$$

With the optimal tuning $\eta = \sqrt{\frac{\ln K}{nK}}$ we have thus shown $\mathbb{E}[\text{Reg}_n] = \mathcal{O}(\sqrt{nK \ln K})$. □

We make two remarks of the proof. First, as we argued earlier, the fact that $\widehat{\ell}_t(a)$ is non-negative is important for the algorithm to work, and this is also reflected in the proof since it is a requirement for Lemma 1 of Lecture 6 to hold. Second, the potentially large variance of the estimator $\mathbb{E}_t[\widehat{\ell}_t^2(a)] = \frac{\ell_t^2(a)}{p_t(a)}$ is automatically canceled by another $p_t(a)$ term in the "stability" term $\sum_{t=1}^n \sum_{a=1}^K p_t(a) \widehat{\ell}_t^2(a)$, which is rather remarkable.

Compared to the full information setting, the regret bound of Exp3 has an extra $\sqrt{K}$ factor, which can be seen as the price of learning with bandit feedback and is in fact unavoidable as we will show soon.

3 Stochastic MAB

Next, we move on to the stochastic setting where the losses for each arm a are i.i.d. samples of a fixed distribution with mean $\mu(a) \in [0, 1]$ and we aim to minimize pseudo regret. Of course, as mentioned

[1] This can be fixed by shifting the loss to $[0, 1]^K$ again.

pseudo regret is bounded by the actual expected regret, and the stochastic setting is just a special case of the adversarial setting, so we can still directly apply Exp3 and get $\overline{\text{Reg}}_n = \mathcal{O}(\sqrt{nK\ln K})$. However, by exploiting the stochasticity we can in fact achieve an even better bound, and perhaps more importantly, this is achieved via a power principle called *optimism in face of the uncertainty*, which is useful for many other problems as well.

Specifically, since the losses are i.i.d. samples with mean $\mu(a)$, it is more than natural to keep track of the empirical mean of arm a up to each time t:

$$\widehat{\mu}_t(a) = \frac{1}{m_t(a)}\sum_{\tau=1}^{t}\mathbf{1}\{a_\tau = a\}\ell_\tau(a) \quad \text{where} \quad m_t(a) = \sum_{\tau=1}^{t}\mathbf{1}\{a_\tau = a\},$$

as an estimate for $\mu(a)$. Intuitively, the more times we select an arm (i.e. larger $m_t(a)$), the better this estimate is. Indeed, one can show the following concentration lemma:

Lemma 1. *No matter what the learner's strategy is, for each arm $a \in [K]$ we have with probability at least $1 - 2/n$,*

$$|\widehat{\mu}_t(a) - \mu(a)| \le 2\sqrt{\frac{\ln n}{m_t(a)}}, \quad \forall t = 1, \ldots, n.$$

The lemma is proven essentially by Hoeffding's inequality, except for the extra technicality needed to deal with the fact that $m_t(a)$ is also random. We omit the proof for simplicity.

With this concentration lemma, at each time we basically have a confidence set of plausible environments. The question is now how the learner should act with this knowledge, and here comes the optimistic principle: *the learner should be optimistic and act as if the environment is the best possible one*. More specifically, based on the concentration lemma the best possible environment using observations up to time $t-1$ is when each $\mu(a)$ is exactly

$$\text{LCB}_t(a) \triangleq \widehat{\mu}_{t-1}(a) - 2\sqrt{\frac{\ln n}{m_{t-1}(a)}}.$$

Therefore, an optimistic learner plays

$$a_t \in \operatorname*{argmin}_{a\in[K]} \text{LCB}_t(a). \tag{1}$$

Here, LCB stands for lower confidence bound. Traditionally this algorithm was derived for the reward (instead of loss) setting and thus optimism means playing an action with the highest upper confidence bound, hence the name UCB algorithm. To be consistent we stick with the loss setting, but for convention we still call the strategy Equation (1) UCB algorithm.

Let's take a closer look at the UCB algorithm. First of all, note that $m_{t-1}(a)$ is initially 0, leading to negative infinity for $\text{LCB}_t(a)$, so the algorithm will be forced to pick each action exactly once for the first K rounds. Afterwards, the two terms in $\text{LCB}_t(a)$ are essentially playing the role of exploitation and exploration respectively, since they suggest picking action with low empirical mean (exploitation) but penalized by how many times it has been selected (exploration). Also note that in contrast to Exp3, UCB is a deterministic algorithm and there is no randomness from the algorithm itself.

We start with a very simple analysis to show that UCB at least achieves something close to the regret bound of Exp3.

Theorem 2. *UCB strategy (1) ensures $\overline{\text{Reg}}_n = \mathcal{O}\left(\sqrt{nK\ln n}\right)$.*

Proof. Conditioning on the event E: $|\widehat{\mu}_t(a) - \mu(a)| \le 2\sqrt{\frac{\ln n}{m_t(a)}}$ holds for all t and a, which happens with probability at least $1 - 2K/n$ by Lemma 1 and a union bound over K arms, we have with

$a^\star \in \operatorname{argmin}_a \mu(a)$,

$$\begin{aligned}
\sum_{t=K+1}^{n} (\mu(a_t) - \mu(a^\star)) &\le \sum_{t=K+1}^{n} (\mu(a_t) - \mathrm{LCB}_t(a^\star)) && \text{(event E)}\\
&\le \sum_{t=K+1}^{n} (\mu(a_t) - \mathrm{LCB}_t(a_t)) && \text{(by Equation (1))}\\
&\le 4\sum_{t=K+1}^{n} \sqrt{\frac{\ln n}{m_{t-1}(a_t)}} && \text{(event E)}\\
&= 4\sum_{t=K+1}^{n}\sum_{a=1}^{K} \mathbf{1}\left\{a = a_t\right\}\sqrt{\frac{\ln n}{m_{t-1}(a)}} = 4\sum_{a=1}^{K}\sum_{t=K+1}^{n} \mathbf{1}\left\{a = a_t\right\}\sqrt{\frac{\ln n}{m_{t-1}(a)}}\\
&= 4\sum_{a=1}^{K}\sum_{s=1}^{m_{n-1}(a)} \sqrt{\frac{\ln n}{s}} \le 8\sum_{a=1}^{K}\sqrt{m_n(a)\ln n}\\
&\le 8\sqrt{\left(\sum_{a=1}^{K} m_n(a)\right) K\ln n} && \text{(Cauchy-Schwarz)}\\
&= 8\sqrt{nK\ln n}.
\end{aligned}$$

Therefore, we have

$$\overline{\mathrm{Reg}}_n \le (K + 8\sqrt{nK\ln n}) + \Pr(\text{E does not hold}) \times n = \mathcal{O}\left(\sqrt{nK\ln n}\right).$$

□

The proof shows that optimism allows us to bound the regret at each time by a deviation term in terms of the selected action $\sqrt{\frac{2\ln n}{m_{t-1}(a_t)}}$, which sums up to $\mathcal{O}\left(\sqrt{nK\ln n}\right)$ by simple calculation.

Of course, as promised we are aiming for an even better bound (since Exp3 achieves a similar bound already). To show the result, we first define the optimality gap of an arm a as

$$\Delta_a = \mu(a) - \mu(a^\star)$$

where $a^\star \in \operatorname{argmin}_a \mu(a)$. Clearly the larger the gap, the worse the arm is. It turns out that UCB achieves a gap-independent regret bound that is only *logarithmic* in n.

Theorem 3. *UCB strategy* (1) *ensures* $\overline{\mathrm{Reg}}_n = \mathcal{O}\left(\sum_{a:\Delta_a>0} \frac{\ln n}{\Delta_a}\right)$.

Proof. We start with rewriting the pseudo regret as

$$\overline{\mathrm{Reg}}_n = \mathbb{E}\left[\sum_{t=1}^{n}(\mu(a_t) - \mu(a^\star))\right] = \mathbb{E}\left[\sum_{t=1}^{n}\sum_{a=1}^{K}\Delta_a \mathbf{1}\{a_t = a\}\right] = \sum_{a:\Delta_a>0}\Delta_a \mathbb{E}\left[m_n(a)\right].$$

It remains to bound $\mathbb{E}\left[m_n(a)\right]$ by $\mathcal{O}\left(\frac{\ln n}{\Delta_a^2}\right)$ for each suboptimal arm a. To show this, let $n_0 = \lceil\frac{16\ln n}{\Delta_a^2}\rceil$ and bound $\mathbb{E}[m_n(a)]$ by

$$n_0 + \sum_{t=n_0+1}^{n} \Pr\left(a_t = a \text{ and } m_{t-1}(a) \ge n_0\right).$$

In the rest of the proof we will show that each summand $\Pr\left(a_t = a \text{ and } m_{t-1}(a) \ge n_0\right)$ is at most $\frac{4}{n}$. Indeed, note that the algorithm picks a suboptimal action a only if one of the following two rare events happens

$$\mathrm{LCB}_t(a^\star) \ge \mu(a^\star) \quad \text{or} \quad \mathrm{LCB}_t(a) \le \mu(a^\star)$$

since otherwise $\mathrm{LCB}_t(a) > \mu(a^\star) > \mathrm{LCB}_t(a^\star)$ and a will not be selected according to the UCB strategy (1). Therefore by a union bound we have that $\Pr\left(a_t = a \text{ and } m_{t-1}(a) \geq n_0\right)$ is bounded by

$$\Pr\left(\mathrm{LCB}_t(a^\star) \geq \mu(a^\star)\right) + \Pr\left(\mathrm{LCB}_t(a) \leq \mu(a^\star) \text{ and } m_{t-1}(a) \geq n_0\right).$$

The first term is bounded by $\frac{2}{n}$ by Lemma 1. For the second term, note that it is equivalent to

$$\begin{aligned} &\Pr\left(\widehat{\mu}_{t-1}(a) - 2\sqrt{\frac{\ln n}{m_{t-1}(a)}} \leq \mu(a^\star) \text{ and } m_{t-1}(a) \geq n_0\right) \\ = \ &\Pr\left(\Delta_a - 2\sqrt{\frac{\ln n}{m_{t-1}(a)}} \leq \mu(a) - \widehat{\mu}_{t-1}(a) \text{ and } m_{t-1}(a) \geq n_0\right) \\ \leq \ &\Pr\left(2\sqrt{\frac{\ln n}{m_{t-1}(a)}} \leq \mu(a) - \widehat{\mu}_{t-1}(a)\right), \qquad \text{(by the choice of } n_0\text{)} \end{aligned}$$

which is also bounded by $\frac{2}{n}$ according to Lemma 1. This shows $\mathbb{E}[m_n(a)] \leq \lceil \frac{16 \ln n}{\Delta_a^2} \rceil + 4$ and thus finishes the proof. □

Again, this gap-dependent bound has only logarithmic dependence on n. If we consider all optimality gaps to be fixed constants, then this bound is much better than the $\sqrt{n}$-type bound for the adversarial case. Even if the gaps are tiny so that this gap-dependent bound is huge, note that we still have a safe guarantee $\overline{\mathrm{Reg}}_n = \mathcal{O}\left(\sqrt{nK \ln n}\right)$ from Theorem 2.

4 Lower Bounds

Finally, we argue that in the worst case, the expected regret of any MAB algorithm is at least $\Omega(\sqrt{nK})$, demonstrating a strict gap between learning with full information and learning with partial information.

The intuition of the lower bound is rather straightforward. For any fixed algorithm, first image running it in a simple world where losses for all arms are generated independently and uniformly from $\{0, 1\}$. There must exist an arm that is selected no more than n/K times by this algorithm. Now suppose we secretly modify the environment so that the loss of this arm follows a Bernoulli distribution with parameter $1/2 - \sqrt{K/n}$, which is not distinguishable from the uniform distribution with only n/K samples by simple arguments from information theory. Then the algorithm should not be aware of this change and will still pick this arm no more than n/K in this new environment, leading to an expected regret $(n - n/K)\sqrt{K/n} = \Omega(\sqrt{nK})$.

The question is how to make this argument formal. In particular, how to formally argue that in the new environment the algorithm's behavior stays roughly the same. As we will see in the proof below, this can in fact be related to the KL divergence between two distributions corresponding to the two environments.

Theorem 4. *For any MAB algorithm $\mathcal{A}$, there exists a fixed sequence of loss vectors such that*

$$\mathbb{E}[\mathrm{Reg}_n] = \Omega(\sqrt{nK}).$$

Proof. According to the informal argument mentioned earlier, we create two randomized environments $\mathcal{E}$ and $\mathcal{E}'$ in the following way (and use $\mathbb{E}, \mathbb{P}$ and $\mathbb{E}', \mathbb{P}'$ to denote the expectation and probability measure in these two environments respectively). In $\mathcal{E}$, every loss $\ell_t(a)$ follows independently a Bernoulli distribution with parameter $1/2$, denoted by $\mathrm{Ber}(1/2)$. There must exist $a' \in [K]$ such that $\mathbb{E}[m(a')] \leq \frac{n}{K}$ where $m(a) = \sum_{t=1}^{n} \mathbf{1}\{a_t = a\}$ is the total number of times a is selected. Then $\mathcal{E}'$ is constructed such that the losses of arm a' follow $\mathrm{Ber}(1/2 - \epsilon)$ independently for some small $\epsilon \leq 1/4$ to be specified later, and every other arms still follow $\mathrm{Ber}(1/2)$ independently.

The rest of the proof argues that $\mathbb{E}'\mathbb{E}_{\mathcal{A}}[\mathrm{Reg}_n] = \Omega(\sqrt{nK})$ where we use $\mathbb{E}_{\mathcal{A}}[\cdot]$ to denote the expectation with respect to the internal randomness of $\mathcal{A}$. This implies that there exists a *fixed* sequence of loss vectors such that $\mathbb{E}_{\mathcal{A}}[\mathrm{Reg}_n] = \Omega(\sqrt{nK})$ and concludes the proof. Further note that $\mathbb{E}'\mathbb{E}_{\mathcal{A}}[\mathrm{Reg}_n] = \mathbb{E}_{\mathcal{A}}\mathbb{E}'[\mathrm{Reg}_n]$, so it is sufficient to prove that for any *deterministic* algorithm, $\mathbb{E}'[\mathrm{Reg}_n] = \Omega(\sqrt{nK})$. If we denote the observation of the learner at time t by $\widetilde{\ell}_t = \ell_t(a_t)$, then a deterministic algorithm selects a_t via some fixed function of $\widetilde{\ell}_{1:t-1}$ (note that the information of $a_{1:t-1}$ is redundant since $a_{1:t-1}$ are determined by $\widetilde{\ell}_{1:t-2}$ already).

Clearly, in expectation a' is the best arm in $\mathcal{E}'$ and

$$\mathbb{E}'[\mathrm{Reg}_n] = \mathbb{E}'\left[\sum_{t=1}^n \ell_t(a_t) - \min_{a\in[K]}\sum_{t=1}^n \ell_t(a)\right] \geq \mathbb{E}'\left[\sum_{t=1}^n \ell_t(a_t) - \sum_{t=1}^n \ell_t(a')\right] = (n - \mathbb{E}'\left[m(a')\right])\epsilon.$$

We next show that $\mathbb{E}'\left[m(a')\right]$ and $\mathbb{E}\left[m(a')\right]$ are close, that is, the number of times a' is selected in environment $\mathcal{E}$ and in environment $\mathcal{E}'$ are similar (just as in the previous informal argument). Indeed,

$$\begin{aligned}\mathbb{E}'\left[m(a')\right] - \mathbb{E}\left[m(a')\right] &= \sum_{\widetilde{\ell}_{1:n}} m(a')\left(\mathbb{P}'(\widetilde{\ell}_{1:n}) - \mathbb{P}(\widetilde{\ell}_{1:n})\right) \leq n\sum_{\widetilde{\ell}_{1:n}}\left|\mathbb{P}'(\widetilde{\ell}_{1:n}) - \mathbb{P}(\widetilde{\ell}_{1:n})\right| \\ &= n\left\|\mathbb{P}' - \mathbb{P}\right\|_1 \leq n\sqrt{2\mathrm{KL}(\mathbb{P}\parallel\mathbb{P}')}.\end{aligned}$$

To calculate $\mathrm{KL}(\mathbb{P}\parallel\mathbb{P}')$, we apply a handy divergence decomposition lemma (Lemma 2, included after this proof):

$$\begin{aligned}\mathrm{KL}(\mathbb{P}\parallel\mathbb{P}') &= \mathbb{E}\left[m(a')\right]\cdot\mathrm{KL}\left(\mathrm{Ber}(1/2)\parallel\mathrm{Ber}(1/2-\epsilon)\right) \\ &= \frac{\mathbb{E}\left[m(a')\right]}{2}\left(\ln\frac{1/2}{1/2+\epsilon} + \ln\frac{1/2}{1/2-\epsilon}\right) \\ &= \frac{\mathbb{E}\left[m(a')\right]}{2}\ln\left(\frac{1}{1-4\epsilon^2}\right) \\ &\leq 4\mathbb{E}\left[m(a')\right]\epsilon^2,\end{aligned}$$

where in the last step we use the fact $\ln\left(\frac{1}{1-x}\right) \leq 2x$ for any $x \leq \frac{1}{2}$. Finally we have

$$\mathbb{E}'\left[m(a')\right] \leq \mathbb{E}\left[m(a')\right] + 2n\epsilon\sqrt{2\mathbb{E}\left[m(a')\right]} \leq \frac{n}{K} + 2n\epsilon\sqrt{\frac{2n}{K}},$$

and thus

$$\mathbb{E}'[\mathrm{Reg}_n] \geq n\left(1 - \frac{1}{K} - 2\epsilon\sqrt{\frac{2n}{K}}\right)\epsilon \geq n\left(\frac{1}{2} - 2\epsilon\sqrt{\frac{2n}{K}}\right)\epsilon$$

Setting $\epsilon = \sqrt{\frac{K}{128n}}$ (to maximize the lower bound above) shows $\mathbb{E}'[\mathrm{Reg}_n] = \Omega(\sqrt{nK})$, finishing the proof. □

The following divergence decomposition lemma is very powerful and is used extensively in proving lower bounds.

Lemma 2 (Divergence decomposition). *Let $\mathcal{E}$ and $\mathcal{E}'$ be two stochastic MAB environments where the losses of arm a are i.i.d. samples of $\mathcal{P}_a$ and $\mathcal{P}'_a$ respectively for each a. Let $\widetilde{\ell}_t = \ell_t(a_t)$ be the observation of a deterministic learner at time t and $\mathbb{P}$ and $\mathbb{P}'$ be the distributions of $\widetilde{\ell}_{1:n}$ for environments $\mathcal{E}$ and $\mathcal{E}'$ respectively. Then*

$$\mathrm{KL}(\mathbb{P}\parallel\mathbb{P}') = \sum_{a=1}^K \mathbb{E}\left[m(a)\right]\mathrm{KL}(\mathcal{P}_a\parallel\mathcal{P}'_a).$$

Proof. For simplicity we consider the case when $\mathcal{P}_{1:K}$ and $\mathcal{P}'_{1:K}$ are discrete distributions (the general case can be proven similarly). By definition and direct calculation we have

$$\begin{aligned}
\mathrm{KL}(\mathbb{P} \parallel \mathbb{P}') &= \sum_{\widetilde{\ell}_{1:n}} \mathbb{P}(\widetilde{\ell}_{1:n}) \ln\left(\frac{\mathbb{P}(\widetilde{\ell}_{1:n})}{\mathbb{P}'(\widetilde{\ell}_{1:n})}\right) = \sum_{\widetilde{\ell}_{1:n}} \mathbb{P}(\widetilde{\ell}_{1:n}) \ln\left(\frac{\prod_{t=1}^{n} \mathbb{P}(\widetilde{\ell}_t|\widetilde{\ell}_{1:t-1})}{\prod_{t=1}^{n} \mathbb{P}'(\widetilde{\ell}_t|\widetilde{\ell}_{1:t-1})}\right) \\
&= \sum_{t=1}^{n} \sum_{\widetilde{\ell}_{1:n}} \mathbb{P}(\widetilde{\ell}_{1:n}) \ln\left(\frac{\mathbb{P}(\widetilde{\ell}_t|\widetilde{\ell}_{1:t-1})}{\mathbb{P}'(\widetilde{\ell}_t|\widetilde{\ell}_{1:t-1})}\right) \\
&= \sum_{t=1}^{n} \sum_{\widetilde{\ell}_{1:t}} \left(\sum_{\widetilde{\ell}_{t+1:n}} \mathbb{P}(\widetilde{\ell}_{t+1:n}|\widetilde{\ell}_{1:t})\right) \mathbb{P}(\widetilde{\ell}_{1:t}) \ln\left(\frac{\mathbb{P}(\widetilde{\ell}_t|\widetilde{\ell}_{1:t-1})}{\mathbb{P}'(\widetilde{\ell}_t|\widetilde{\ell}_{1:t-1})}\right) \\
&= \sum_{t=1}^{n} \sum_{\widetilde{\ell}_{1:t}} \mathbb{P}(\widetilde{\ell}_{1:t}) \ln\left(\frac{\mathbb{P}(\widetilde{\ell}_t|\widetilde{\ell}_{1:t-1})}{\mathbb{P}'(\widetilde{\ell}_t|\widetilde{\ell}_{1:t-1})}\right) \\
&= \sum_{a=1}^{K} \sum_{t=1}^{n} \sum_{\widetilde{\ell}_{1:t}:a_t=a} \mathbb{P}(\widetilde{\ell}_{1:t}) \ln\left(\frac{\mathbb{P}(\widetilde{\ell}_t|\widetilde{\ell}_{1:t-1})}{\mathbb{P}'(\widetilde{\ell}_t|\widetilde{\ell}_{1:t-1})}\right) \\
&= \sum_{a=1}^{K} \sum_{t=1}^{T} \sum_{\widetilde{\ell}_{1:t-1}:a_t=a} \mathbb{P}(\widetilde{\ell}_{1:t-1}) \sum_{\widetilde{\ell}_t} \mathbb{P}(\widetilde{\ell}_t|\widetilde{\ell}_{1:t-1}) \ln\left(\frac{\mathbb{P}(\widetilde{\ell}_t|\widetilde{\ell}_{1:t-1})}{\mathbb{P}'(\widetilde{\ell}_t|\widetilde{\ell}_{1:t-1})}\right) \\
&= \sum_{a=1}^{K} \sum_{t=1}^{T} \mathbb{P}(a_t = a)\mathrm{KL}(\mathcal{P}_a \parallel \mathcal{P}'_a) = \sum_{a=1}^{K} \mathbb{E}\left[m(a)\right] \mathrm{KL}(\mathcal{P}_a \parallel \mathcal{P}'_a),
\end{aligned}$$

fishing the proof. □