CSCI 678: Theoretical Machine Learning
Lecture 8

Fall 2024, Instructor: Haipeng Luo

1 From Values to Algorithms — A General Recipe

In the last two lectures, we discussed several online learning algorithms. One might wonder: how to come up with an algorithm, especially when facing a new problem? Is there a general principle or even a concrete recipe to design algorithms?

To answer this question, we come back to the general setup and its minimax formulation:

$$\mathcal{V}^{\text{seq}}(\mathcal{F}, n) = \left\langle\!\!\left\langle \inf_{q_t \in \Delta(\mathcal{D})} \sup_{z_t \in \mathcal{Z}} \mathbb{E}_{\widehat{y}_t \sim q_t} \right\rangle\!\!\right\rangle_{t=1}^{n} \left[\frac{\text{Reg}(\mathcal{F}, n)}{n} \right].$$

Previously, we derived a sequence of upper bounds on this minimax expression to study learnability, without having any concrete algorithms. However, in principle, one can be ambiguous and ask for the *exact minimax optimal* algorithm, which at time t predicts $\widehat{y}_t$ drawn from

$$\begin{aligned}
&\operatorname*{argmin}_{q_t \in \Delta(\mathcal{D})} \sup_{z_t \in \mathcal{Z}} \mathbb{E}_{\widehat{y}_t \sim q_t} \left[\left\langle\!\!\left\langle \inf_{q_\tau \in \Delta(\mathcal{D})} \sup_{z_\tau \in \mathcal{Z}} \mathbb{E}_{\widehat{y}_\tau \sim q_\tau} \right\rangle\!\!\right\rangle_{\tau=t+1}^{n} \text{Reg}(\mathcal{F}, n) \right] \\
&= \operatorname*{argmin}_{q_t \in \Delta(\mathcal{D})} \sup_{z_t \in \mathcal{Z}} \mathbb{E}_{\widehat{y}_t \sim q_t} \left[\left\langle\!\!\left\langle \inf_{q_\tau \in \Delta(\mathcal{D})} \sup_{z_\tau \in \mathcal{Z}} \mathbb{E}_{\widehat{y}_\tau \sim q_\tau} \right\rangle\!\!\right\rangle_{\tau=t+1}^{n} \left[\sum_{s=t}^{n} \ell(\widehat{y}_s, z_s) - \inf_{f \in \mathcal{F}} \sum_{s=1}^{n} \ell(f, z_s) \right] \right] \\
&= \operatorname*{argmin}_{q_t \in \Delta(\mathcal{D})} \sup_{z_t \in \mathcal{Z}} \left(\mathbb{E}_{\widehat{y}_t \sim q_t} \left[\ell(\widehat{y}_t, z_t) \right] + \left\langle\!\!\left\langle \inf_{q_\tau \in \Delta(\mathcal{D})} \sup_{z_\tau \in \mathcal{Z}} \mathbb{E}_{\widehat{y}_\tau \sim q_\tau} \right\rangle\!\!\right\rangle_{\tau=t+1}^{n} \left[\sum_{s=t+1}^{n} \ell(\widehat{y}_s, z_s) - \inf_{f \in \mathcal{F}} \sum_{s=1}^{n} \ell(f, z_s) \right] \right).
\end{aligned}$$

To simplify notation, recursively define the (unnormalized) *conditional value* of the game given the past decisions $z_{1:t}$ of the environment as

$$\begin{aligned}
\mathcal{V}_n(z_{1:t}) &= \inf_{q \in \Delta(\mathcal{D})} \sup_{z \in \mathcal{Z}} \left(\mathbb{E}_{\widehat{y} \sim q} \left[\ell(\widehat{y}, z) \right] + \mathcal{V}_n(z_{1:t}, z) \right) \\
\text{with} \qquad \mathcal{V}_n(z_{1:n}) &= - \inf_{f \in \mathcal{F}} \sum_{t=1}^{n} \ell(f, z_t).
\end{aligned} \tag{1}$$

In words, $\mathcal{V}_n(z_{1:t})$ is the optimal regret (offset by the loss already suffered) one can achieve against the worst-case future, given the past t steps. Clearly, we have $\mathcal{V}^{\text{seq}}(\mathcal{F}, n) = \frac{1}{n}\mathcal{V}_n(\emptyset)$. With this conditional value, the minimax strategy becomes

$$q_t = \operatorname*{argmin}_{q \in \Delta(\mathcal{D})} \sup_{z \in \mathcal{Z}} \left(\mathbb{E}_{\widehat{y} \sim q} \left[\ell(\widehat{y}, z) \right] + \mathcal{V}_n(z_{1:t-1}, z) \right).$$

In principle, finding q_t is a dynamic program. For most cases, however, there is no simple closed-form solution or efficient way to solve it exactly. Instead, we aim for finding an approximate solution. One general way to do so is to search for a *relaxation* of the conditional value. A relaxation Rel_n is a sequence of functions that map the past decisions of the environment $z_{1:t}$ to a real

value for each $t = 1, \dots, n$ (just like the conditional value $\mathcal{V}_n$). It is called *admissible* if for any $z_1, \dots, z_T \in \mathcal{Z}$,

$$\begin{aligned} &\forall t = 0, \dots, n-1, \quad \mathrm{Rel}_n(z_{1:t}) \geq \inf_{q \in \Delta(\mathcal{D})} \sup_{z \in \mathcal{Z}} \left(\mathbb{E}_{\widehat{y} \sim q} \left[\ell(\widehat{y}, z) \right] + \mathrm{Rel}_n(z_{1:t}, z) \right) \\ &\text{and} \qquad \mathrm{Rel}_n(z_{1:n}) \geq - \inf_{f \in \mathcal{F}} \sum_{t=1}^{n} \ell(f, z_t). \end{aligned} \tag{2}$$

Comparing Equation (1) and Equation (2), it is clear that $\mathcal{V}_n(z_{1:t}) \leq \mathrm{Rel}_n(z_{1:t})$ always holds, that is, admissible relaxation is indeed an upper bound of the conditional value. More importantly, if we replace the conditional value by the relaxation in the minimax optimal strategy, that is,

$$q_t = \operatorname*{argmin}_{q \in \Delta(\mathcal{D})} \sup_{z \in \mathcal{Z}} \left(\mathbb{E}_{\widehat{y} \sim q} \left[\ell(\widehat{y}, z) \right] + \mathrm{Rel}_n(z_{1:t-1}, z) \right), \tag{3}$$

then the regret of this algorithm is bounded by $\mathrm{Rel}_n(\emptyset)$. In fact, we do not even need to solve Equation (3) exactly — as long as q_t is such that

$$\sup_{z \in \mathcal{Z}} \left(\mathbb{E}_{\widehat{y} \sim q_t} \left[\ell(\widehat{y}, z) \right] + \mathrm{Rel}_n(z_{1:t-1}, z) \right) \leq \mathrm{Rel}_n(z_{1:t-1}), \tag{4}$$

the regret is bounded by $\mathrm{Rel}_n(\emptyset)$ (note that the solution of Equation (3) always satisfies the above by admissibility). To see this, we just need to repeatedly peel off each term in the regret:

$$\begin{aligned} \mathbb{E}\left[\mathrm{Reg}(\mathcal{F}, n) \right] &\leq \mathbb{E}\left[\sum_{t=1}^{n} \ell(\widehat{y}_t, z_t) + \mathrm{Rel}_n(z_{1:n}) \right] \\ &\leq \mathbb{E}\left[\sum_{t=1}^{n-1} \ell(\widehat{y}_t, z_t) + \sup_{z} \left(\mathbb{E}_{\widehat{y} \sim q_n} \left[\ell(\widehat{y}, z) \right] + \mathrm{Rel}_n(z_{1:n-1}, z) \right) \right] \\ &\leq \mathbb{E}\left[\sum_{t=1}^{n-1} \ell(\widehat{y}_t, z_t) + \mathrm{Rel}_n(z_{1:n-1}) \right] \leq \cdots \leq \mathrm{Rel}_n(\emptyset). \end{aligned}$$

Therefore, as long as we can find an admissible relaxation that is easy to compute and that is not too large, we have a reasonable algorithm. But how to we find such a good relaxation? In fact, in some sense we have seen one already while discussing learnability. Recall that the value of the game is bounded by (twice) the sequential Rademacher complexity, which after scaling means $\mathcal{V}_n(\emptyset) \leq \sup_{\boldsymbol{z}} \mathbb{E}_{\boldsymbol{\epsilon}} \left[\sup_{f \in \mathcal{F}} 2 \sum_{t=1}^{n} \epsilon_t \ell(f, \boldsymbol{z}_t(\boldsymbol{\epsilon})) \right]$. We generalize the concept of sequential Rademacher complexity and define:

$$\mathcal{R}_n(z_{1:t}) = \sup_{\boldsymbol{z}} \mathbb{E}_{\epsilon_{t+1:n}} \sup_{f \in \mathcal{F}} \left(2 \sum_{s=t+1}^{n} \epsilon_s \ell(f, \boldsymbol{z}_{s-t}(\epsilon_{t+1:s-1})) - \sum_{s=1}^{t} \ell(f, z_s) \right), \tag{5}$$

where $\boldsymbol{z}$ ranges over all $\mathcal{Z}$-valued tree with depth $n - t$, which represents the worst-case future. It can be shown that sequential Rademacher complexity is indeed an admissible relaxation.

Theorem 1. *The generalized sequential Rademacher complexity Equation (5) is an admissible relaxation.*

The proof is based on the same symmetrization technique we have discussed before. Roughly speaking, the conditional value given $z_{1:t}$ is in terms of the difference between the total future loss of the learner and the benchmark. The first t terms of the benchmark appear as the second summation in Equation (5), while the last $n - t$ terms is combined with the future total loss of the learner via symmetrization to become the first summation of Equation (5). We leave the complete proof as an exercise.

With this relaxation, we can assert that the strategy defined in Equation (4) is definitely a good algorithm with low regret — as mentioned the regret is bounded as $\mathrm{Rel}_n(\emptyset) = \mathcal{R}_n(\emptyset)$, which is exactly the original sequential Rademacher complexity (scaled by n) and is very close to the value of the game according to previous lectures. However, is there a way to efficiently compute this relaxation? Unfortunately, the answer is still no usually, especially due to the part $\sup_{\boldsymbol{z}}$. Nevertheless, this relaxation is already much manageable compared to the conditional value, and very often, further bounding this relaxation via simple algebra will lead to another relaxation that is efficiently

computable and at the same time small enough. This gives a general "recipe" to design an online learning algorithm:

1. Start with the sequential Rademacher complexity Equation (5).
2. Derive an upper bound of it to get a relaxation that is easy to compute.
3. Prove that the relaxation is admissible.
4. Derive the final algorithm using Equation (3) or Equation (4).

Figure 1: A general recipe to derive an online learning algorithm

Let's now see a concrete example. Consider the case when the loss function is linear: $\ell(f, z) = \langle f, z \rangle$, which, as discussed last time, is representative for all convex losses. Further restrict our attention to $\mathcal{F} = \mathcal{Z} = B_2^d$, the unit ℓ_2 ball.

Step 1. The generalized sequential Rademacher complexity in this case reduces to

$$\begin{aligned}\mathcal{R}_n(z_{1:t}) &= \sup_{\boldsymbol{z}} \mathbb{E}_{\epsilon_{t+1:n}} \sup_{f \in B_2^d} \left\langle f, 2 \sum_{s=t+1}^{n} \epsilon_s \boldsymbol{z}_{s-t}(\epsilon_{t+1:s-1}) - \sum_{s=1}^{t} z_s \right\rangle \\ &= \sup_{\boldsymbol{z}} \mathbb{E}_{\epsilon_{t+1:n}} \left\| 2 \sum_{s=t+1}^{n} \epsilon_s \boldsymbol{z}_{s-t}(\epsilon_{t+1:s-1}) - Z_t \right\|_2 . \qquad \text{(define } Z_t = \textstyle\sum_{s=1}^{t} z_s\text{)}\end{aligned}$$

Step 2. Using Jensen's equality, we upper bound it as (similarly to Question (a) of HW1)

$$\begin{aligned}\mathcal{R}_n(z_{1:t}) &\le \sup_{\boldsymbol{z}} \sqrt{\mathbb{E}_{\epsilon_{t+1:n}} \left\| 2 \sum_{s=t+1}^{n} \epsilon_s \boldsymbol{z}_{s-t}(\epsilon_{t+1:s-1}) - Z_t \right\|_2^2} \\ &= \sup_{\boldsymbol{z}} \sqrt{4\mathbb{E}_{\epsilon_{t+1:n}} \left[\sum_{s=t+1}^{n} \|\boldsymbol{z}_{s-t}(\epsilon_{t+1:s-1})\|_2^2 \right] + \|Z_t\|_2^2} \\ &\qquad \text{(all other terms have zero mean)} \\ &\le \sqrt{4(n-t) + \|Z_t\|_2^2},\end{aligned}$$

and take the last simple expression as the relaxation: $\mathrm{Rel}_n(z_{1:t}) = \sqrt{4(n-t) + \|Z_t\|_2^2}$.

Step 3. Next, we prove that this relaxation is indeed admissible. The second line of Equation (2) holds with equality. To show the first line, we consider a deterministic and proper strategy such that

$$\inf_{q \in \Delta(\mathcal{D})} \sup_{z \in \mathcal{Z}} \left(\mathbb{E}_{\widehat{y} \sim q} \left[\ell(\widehat{y}, z) \right] + \mathrm{Rel}_n(z_{1:t}, z) \right) \le \inf_{\widehat{y} \in B_2^d} \sup_{z \in B_2^d} \left(\langle \widehat{y}, z \rangle + \sqrt{4(n-t-1) + \|Z_t + z\|_2^2} \right)$$

$$\le \inf_{\widehat{y} \in B_2^d} \sup_{z \in B_2^d} \left(\langle \widehat{y}, z \rangle + \sqrt{4(n-t) + \|Z_t\|_2^2 + 2\langle Z_t, z \rangle} \right).$$

Note that the optimal $\widehat{y}$ must be collinear with Z_t, for otherwise, $\widehat{y}$ has some component that is perpendicular to Z_t, and the environment could potentially add a component in the same direction to increase the term $\langle \widehat{y}, z \rangle$, while keeping the rest unchanged. Therefore, it is best for $\widehat{y}$ to have no component perpendicular to Z_t. Furthermore, it is also easy to see that the optimal $\widehat{y}$ must be in the opposite direction of Z_t (try to convince yourself), meaning that the optimal $\widehat{y}$ can be written as $-\alpha Z_t$ for some coefficient $\alpha \in \mathcal{A} = (0, 1/\|Z_t\|_2]$ (such that $\widehat{y} \in B_2^d$). Using this form we arrive at

$$\inf_{q \in \Delta(\mathcal{D})} \sup_{z \in \mathcal{Z}} \left(\mathbb{E}_{\widehat{y} \sim q} \left[\ell(\widehat{y}, z) \right] + \mathrm{Rel}_n(z_{1:t}, z) \right)$$

$$
\begin{aligned}
&\le \inf_{\alpha\in\mathcal{A}} \sup_{z\in B_2^d} \left(-\alpha \langle Z_t, z\rangle + \sqrt{4(n-t) + \|Z_t\|_2^2 + 2\langle Z_t, z\rangle}\right) \\
&\le \inf_{\alpha\in\mathcal{A}} \sup_{\beta\in\mathbb{R}} \left(-\alpha\beta + \sqrt{4(n-t) + \|Z_t\|_2^2 + 2\beta}\right) && \text{(replacing } \langle Z_t, z\rangle \text{ with } \beta) \\
&= \inf_{\alpha\in\mathcal{A}} \frac{1}{2}\left(\frac{1}{\alpha} + \alpha\left(4(n-t) + \|Z_t\|_2^2\right)\right) && \text{(optimal } \beta \text{ is } \tfrac{1}{2\alpha^2} - 2(n-t) - \tfrac{1}{2}\|Z_t\|_2^2) \\
&= \sqrt{4(n-t) + \|Z_t\|_2^2} = \mathcal{R}_n(z_{1:t}). && \text{(optimal } \alpha \text{ is } \tfrac{1}{\sqrt{4(n-t)+\|Z_t\|_2^2}})
\end{aligned}
$$

This shows that the relaxation is indeed admissible.

Step 4. In fact, the derivation above also implies that the following algorithm satisfies Equation (4):

$$
\widehat{y}_{t+1} = -\alpha_t \sum_{s=1}^{t} z_s, \quad \text{where} \quad \alpha_t = \frac{1}{\sqrt{4(n-t) + \|Z_t\|_2^2}},
$$

and we know that its regret is bounded by $\mathrm{Rel}_n(\emptyset) = 2\sqrt{n}$. Note that this is very close to the gradient descent algorithm discussed last time:

$$
\widehat{y}_{t+1} = \operatorname*{argmin}_{\widehat{y}\in B_2^d} \|\widehat{y} + \eta Z_t\|_2^2 = \begin{cases} -\eta Z_t, & \text{if } \eta\|Z_t\|_2 < 1, \\ \frac{-Z_t}{\|Z_t\|_2}, & \text{else.} \end{cases}
$$

While both algorithms enjoy $\mathcal{O}(\sqrt{n})$ regret, the former is slightly simpler without any explicit projection or learning rate.

Summary. This is just one simple example to illustrate the power of the general recipe. Using this framework one can in fact recover many other algorithms, such as FTRL (and hence Hedge as well), and also derive others that were not known before. This shows that existing algorithms are not "methods that just work" — they can in fact all be derived in a principled way, starting from the fundamental sequential Rademacher complexity.

2 Multi-Armed Bandits

At this point, we have finished all discussions on the learnability and concrete algorithms for online learning problems with "full information", where full information refers to the fact 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 the loss they would have suffered if a different action $\widehat{y}$ would have been taken, by simply evaluating $\ell(\widehat{y}, z_t)$.

As discussed in Lecture 1, while this full-information model captures many problems, there is also a wide range of problems where z_t is not completely revealed and the learner is required to learn under partial information or limited feedback. In the rest of this course, we focus on some canonical 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 $\mathcal{D} = \mathcal{F} = \{1, \dots, K\} \triangleq [K]$ 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 and describe the problem equivalently as follows: The environment first decides n loss vectors: $\ell_1, \dots, \ell_n \in [0,1]^K$ (knowing the learner's algorithm). Then for each $t = 1, \dots, n$, 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 they 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 canonical 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 arms 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 that as usual, the goal of the learner is to minimize regret, defined as

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

where we abbreviate $\mathrm{Reg}(\mathcal{F}, n)$ as Reg_n since $\mathcal{F}$ is now fixed to $[K]$. So far we have made no assumption on how the loss vectors $\ell_1, \dots, \ell_n$ 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 $\ell_1(a), \dots, \ell_n(a)$ 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 $\mathbb{E}[\mathrm{Reg}_n]$, the difference is that we push the expectation inside the "min", which also means $\overline{\mathrm{Reg}}_n \leq \mathbb{E}[\mathrm{Reg}_n]$. Dealing with pseudo regret allows us the ignore the deviation of the samples $\ell_1(a), \dots, \ell_n(a)$ 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.

3 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 might seem impossible to do so in the adversarial setting given that ℓ_t is completely arbitrary; indeed, how can seeing one coordinate of an arbitrary vector tell us anything about the rest of it? It turns out that, however, randomness can help here. Specifically, if we choose a_t randomly according to a fully-supported distribution such that the probability of selecting arm a is $p_t(a) > 0$, then we can construct the *importance-weighted estimator*, 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]

We now analyze the regret of Exp3. It might be tempting to conclude that, just like Hedge, Exp3 achieves regret $\mathbb{E}[\mathrm{Reg}_n] = \mathcal{O}(\sqrt{n \ln K})$ 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, that is, Lemma 1 of Lecture 6, included below again with the generic loss vector ℓ_t renamed as $\widehat{\ell}_t$ for ease of reading, reveals that its regret is $\mathcal{O}(C\sqrt{n \ln K})$ for losses bounded by $C > 0$.

Lemma 1. *For any $\widehat{\ell}_t \in \mathbb{R}^K$ and $\eta > 0$ such that $\eta\widehat{\ell}_t(a) \geq -1$ for all t and a, define $p_t \in \Delta(K)$ to be a distribution such that $p_t(a) \propto \exp\left(-\eta \sum_{\tau=1}^{t-1} \widehat{\ell}_\tau(a)\right)$. Then we have for any $a^\star \in [K]$,*

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

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

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

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

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

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 thus taking expectation on both sides 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 $\mathbb{E}[\mathrm{Reg}_n] = \mathcal{O}(\sqrt{nK \ln K})$. □

We make two remarks for this 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 makes sure that the condition $\eta\widehat{\ell}_t(i) \geq -1$ of Lemma 1 holds. 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 $\sum_{t=1}^n \sum_{a=1}^K p_t(a)\widehat{\ell}_t^2(a)$, 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. (The $\sqrt{\ln K}$ factor in the regret bound, however, is unnecessary and can be removed by using different algorithms.)

4 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, 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 $\overline{\mathrm{Reg}}_n = \mathcal{O}(\sqrt{nK \ln K})$. However, by exploiting the stochasticity we can in fact achieve an even better bound, and perhaps more importantly, this is achieved via a general optimistic principle that 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 often we select an arm (that is, the larger $m_t(a)$ is), the better this estimate is. Indeed, one can show the following concentration lemma:

Lemma 2. *No matter what the learner's strategy is, we have with probability at least $1 - 2K/n$, for every arm $a \in [K]$ and every round $t = 1, \ldots, n$:*

$$|\widehat{\mu}_t(a) - \mu(a)| \leq 2\sqrt{\frac{\ln n}{m_t(a)}}.$$

This lemma can be proven by the standard 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. Note that this lemma captures the even more intuitive tension between exploration and exploitation in stochastic MAB — on the one hand, we want to exploit by picking the empirically best action $\operatorname{argmin}_a \widehat{\mu}_t(a)$, but on the other hand, we also need to explore so that all actions are picked frequently enough to make sure that $\widehat{\mu}_t(a)$ is indeed a good approximation of $\mu(a)$. So how should we balance exploration and exploitation in this case?

Naive Exploration. Based on the intuition above, one naive approach would be "Explore-then-Exploit": first spend Kn_0 (for some parameter n_0) rounds for *uniform exploration*, that is, pick every action for exactly n_0 times; then, in all remaining rounds, exploit by always picking the empirically best action based on the exploration data, that is, $a_t \in \operatorname{argmin}_{a\in[K]} \widehat{\mu}_{Kn_0}(a)$ for all $t > Kn_0$.

While simple, this approach is also intuitively wasteful due to its uniform exploration: every action is selected equally often in the exploration phase even if some of them look much worse than others. A quick analysis shows that such naive exploration indeed leads to suboptimal regret.

Theorem 3. *The Explore-then-Exploit algorithm described above achieves*

$$\overline{\mathrm{Reg}}_n \le Kn_0 + 4n\sqrt{\frac{\ln n}{n_0}} + 2K,$$

which is of order $\mathcal{O}\left(n^{\frac{2}{3}}(K\ln n)^{\frac{1}{3}}\right)$ *after picking the optimal* n_0.

Proof. Let E denote the event stated in Lemma 2, which happens with probability at least $1-2K/n$. Under this event, we know that after the exploration phase, we have for all $a \in [K]$,

$$|\widehat{\mu}_{Kn_0}(a) - \mu(a)| \le 2\sqrt{\frac{\ln n}{n_0}}.$$

Therefore, in the exploitation phase, the loss of the selected action $a_t \in \operatorname{argmin}_{a\in[K]} \widehat{\mu}_{Kn_0}(a)$ is close to that of the optimal action $a^\star \in \operatorname{argmin}_{a\in[K]} \mu(a)$ in the following sense:

$$\mu(a_t) - \mu(a^\star) \le \widehat{\mu}_{Kn_0}(a_t) - \widehat{\mu}_{Kn_0}(a^\star) + 4\sqrt{\frac{\ln n}{n_0}} \le 4\sqrt{\frac{\ln n}{n_0}}.$$

Consequently, under event E, we have

$$\begin{aligned}\sum_{t=1}^{n}(\mu(a_t)-\mu(a^\star)) &= \sum_{t=1}^{Kn_0}(\mu(a_t)-\mu(a^\star)) + \sum_{t=Kn_0+1}^{n}(\mu(a_t)-\mu(a^\star)) \\ &\le (K-1)n_0 + (n-Kn_0)\cdot 4\sqrt{\frac{\ln n}{n_0}} \le Kn_0 + 4n\sqrt{\frac{\ln n}{n_0}}.\end{aligned}$$

Further bounding the regret trivially by n when the event E does not hold, which happens with probability at most $2K/n$, proves the claimed pseudo regret bound. □

Such a regret bound is suboptimal since, as mentioned, applying Exp3 directly already ensures $\overline{\mathrm{Reg}}_n = \mathcal{O}(\sqrt{nK\ln K})$. To improve the algorithm, we need a better and more adaptive exploration scheme. In the next lecture, we will discuss a simple method to achieve so.