CSCI 678: Theoretical Machine Learning
Lecture 9

Fall 2026, Instructor: Haipeng Luo

1 Adaptive Exploration for Stochastic MAB

Recall the stochastic K-armed bandit problem discussed last time: for each time $t = 1, \ldots, n$, the loss $\ell_t(a) \in [0, 1]$ for each arm a is independently drawn from a fixed distribution with mean $\mu(a)$; the learner selects one of the K arms $a_t \in [K]$ and observes only $\ell_t(a_t)$. The goal of the learner is to minimize her pseudo regret, defined as

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

where $a^\star \in \operatorname{argmin}_{a \in [K]} \sum_{t=1}^{n} \mu(a)$ is an optimal arm.

In the last lecture, we saw that the naive Explore-then-Exploit approach achieves suboptimal regret due to its uniform and nonadaptive exploration. Now, we introduce a more adaptive exploration scheme based on a general principle called "optimism in face of uncertainty", which is useful for many other problems with partial information. The main idea is the following: among all plausible environments that are consistent with the data observed so far, *the learner should be optimistic and act as if the environment is the best possible one*. To see what this means and how to apply it to our problem, first recall the concentration lemma discussed last time:

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

Here, $m_t(a) = \sum_{\tau=1}^{t} \mathbf{1}\{a_\tau = a\}$ is the number of times arm a has been pulled up to round t and $\widehat{\mu}_t(a) = \frac{1}{m_t(a)} \sum_{\tau=1}^{t} \mathbf{1}\{a_\tau = a\}\ell_\tau(a)$ is the empirical average of the observed losses for arm a up to round t. This lemma tells us exactly what the plausible environments are given the data observed so far. Among them, the most favorable one to the learner is the one with the smallest expected loss, that is, when the mean of the loss for each action a is the following lower confidence bound (LCB):

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

Therefore, an optimistic learner would wishfully believe that this is the true environment and naturally selects

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

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. By convention, we use the same name (even though technically it really should be called the LCB algorithm).

A couple of remarks on the UCB algorithm are in order. First, note that $m_{t-1}(a)$ is initially 0, leading to negative infinity for $\mathrm{LCB}_t(a)$, so the algorithm will be forced to pick each action exactly once for the first K rounds. Afterwards, the two terms in $\mathrm{LCB}_t(a)$ are essentially playing the role of exploitation and exploration respectively: the first term suggests picking actions with low empirical mean (exploitation), while the second term suggests picking actions that have not been selected often enough (exploration). Together, they achieve an adaptive trade-off between exploitation and exploration, which does not waste many rounds to explore an infrequently selected action if it already looks pretty bad. Also note that optimism is indeed important to derive exploration — think about what would happen if we instead adopt pessimism and pick the arm with the smallest upper confidence bound $\widehat{\mu}_{t-1}(a) + 2\sqrt{(\ln n)/m_{t-1}(a)}$. Moreover, in contrast to Exp3, UCB is a deterministic algorithm, that is, no randomness is used in deciding which actions to play.

To analyze the UCB algorithm, we start by rewriting the pseudo regret as follows:

$$\overline{\mathrm{Reg}}_n = \mathbb{E}\left[\sum_{t=1}^{n} \mu(a_t) - \sum_{t=1}^{n} \mu(a^\star)\right] = \mathbb{E}\left[\sum_{t=1}^{n}\sum_{a=1}^{K} \Delta_a \mathbf{1}\{a_t = a\}\right] = \sum_{a:\Delta_a>0} \Delta_a \mathbb{E}[m_n(a)] \quad (2)$$

where $\Delta_a = \mu(a) - \mu(a^\star)$ is the *suboptimality gap* of action a. For the UCB algorithm, the suboptimality gap of an action and the number of times it has been pulled can be connected using the following key lemma.

Lemma 2. *Under the event stated in Lemma 1, UCB ensures* $\Delta_{a_t} \le 4\sqrt{\frac{\ln n}{m_{t-1}(a_t)}}$ *for any* $t = 1, \ldots, n$.

Proof. This is a direct consequence of optimism:

$$\begin{aligned}
\Delta_{a_t} = \mu(a_t) - \mu(a^\star) &\le \mu(a_t) - \mathrm{LCB}_t(a^\star) && \text{(Lemma 1)} \\
&\le \mu(a_t) - \mathrm{LCB}_t(a_t) && \text{(since } a_t \text{ has the smallest LCB)} \\
&\le 4\sqrt{\tfrac{\ln n}{m_{t-1}(a_t)}}, && \text{(by the definition of LCB and Lemma 1)}
\end{aligned}$$

which completes the proof. □

This simple fact immediately tells us in total how many times each action can be selected by UCB, which also leads to a gap-dependent logarithmic (in n) pseudo regret bound.

Theorem 1. *Under the event stated in Lemma 1, we have* $m_n(a) \le \frac{16 \ln n}{\Delta_a^2} + 1$ *for every action* a. *Consequently, the pseudo-regret of UCB satisfies* $\overline{\mathrm{Reg}}_n = \mathcal{O}(\sum_{a:\Delta_a>0} \frac{\ln n}{\Delta_a})$.

Proof. For each action a, applying Lemma 2 with t being the last time a is selected shows: $\Delta_a \le 4\sqrt{\frac{\ln n}{m_n(a)-1}}$, which, after rearranging, proves the first statement. Therefore, if we denote the event stated in Lemma 1 by E, then the pseudo regret (recall Eq. (2)) of UCB can be bounded as:

$$\begin{aligned}
\overline{\mathrm{Reg}}_n &\le \Pr(E) \times \sum_{a:\Delta_a>0} \Delta_a \mathbb{E}\left[m_n(a) \mid E\right] + \Pr(\neg E) \times n \\
&\le \sum_{a:\Delta_a>0} \left(\frac{16 \ln n}{\Delta_a} + \Delta_a\right) + 2K = \mathcal{O}\left(\sum_{a:\Delta_a>0} \frac{\ln n}{\Delta_a}\right),
\end{aligned}$$

which proves the second statement. □

Note that this is an *instance-dependent* regret bound: it depends on the suboptimality gaps of a particular instance. Treating these gaps as constants, we see that this bound enjoys logarithmic dependence on n, which is an exponential improvement compared to the $\mathcal{O}(\sqrt{nK \ln K})$ bound achieved by Exp3.

In this instance-optimal bound, the smaller the gaps, the larger the pseudo regret, which makes sense to some degree because smaller gaps make it harder to distinguish the optimal actions from the rest. On the other hand, however, if an action really has a tiny suboptimality gap, then by definition

selecting it does not lead to large regret and thus there is really no point in distinguishing it from the optimal actions. Building on this intuition, in HW4 you will further prove that UCB indeed also guarantees $\mathcal{O}(\sqrt{nK \ln n})$ pseudo regret at the same time, no matter how small the gaps are.

To conclude, by exploiting the stochasticity of the problem and applying the optimism in face of uncertainty principle, the UCB algorithm achieves a more adaptive and efficient trade-off between exploration and exploitation, resulting in a regret guarantee that is never worse than the naive Explore-then-Exploit algorithm or even Exp3 (ignoring logarithmic factors), and is potentially much better when the suboptimality gaps are not too small.

2 Lower Bounds for MAB

Recall that with full information, learning with K actions using the Hedge algorithm suffers at most $\mathcal{O}(\sqrt{n \ln K})$ regret. On the other hand, when learning with partial information, the regret bound of Exp3 exhibits an extra factor of $\sqrt{K}$, which is quite intuitive since each round we only obtain $1/K$ fraction of information. In this section, we prove that such increase in the regret is essentially unavoidable. More specifically, we will 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 imagine 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 that the adversary secretly modifies 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 based on standard arguments from information theory. Thus, when run in this new environment, the same algorithm should not be aware of this change and will still pick this arm not often enough, say no more than $n/2$ rounds, leading to at least $\frac{n}{2}\sqrt{K/n} = \Omega(\sqrt{nK})$ regret.

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 2. *For any MAB algorithm $\mathcal{A}$ and $n \geq K$, there exists a fixed sequence of loss vectors such that*

$$\mathbb{E}_{\mathcal{A}}[\text{Reg}_n] = \Omega(\sqrt{nK}),$$

where $\mathbb{E}_{\mathcal{A}}[\cdots]$ denotes the expectation with respect to the randomness of $\mathcal{A}$.

Proof. According to the informal argument mentioned earlier, we create two randomized environments $\mathcal{E}$ and $\mathcal{E}'$ in the following way (and use $\mathbb{E}$ and $\mathbb{E}'$ to denote the expectation in these two environments respectively). In $\mathcal{E}$, every loss $\ell_t(a)$ follows independently a Bernoulli distribution with parameter $1/2$, denoted by $\text{Ber}(1/2)$. 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 $\text{Ber}(1/2 - \epsilon)$ independently for some small $\epsilon \leq 1/4$ to be specified later, and all other arms still follow $\text{Ber}(1/2)$ independently.

The rest of the proof argues that $\mathbb{E}'\mathbb{E}_{\mathcal{A}}[\text{Reg}_n] = \Omega(\sqrt{nK})$, which implies that there exists a *fixed* sequence of loss vectors such that $\mathbb{E}_{\mathcal{A}}[\text{Reg}_n] = \Omega(\sqrt{nK})$ and concludes the proof. Further note that $\mathbb{E}'\mathbb{E}_{\mathcal{A}}[\text{Reg}_n] = \mathbb{E}_{\mathcal{A}}\mathbb{E}'[\text{Reg}_n]$, so it is sufficient to prove that for any *deterministic* algorithm, $\mathbb{E}'[\text{Reg}_n] = \Omega(\sqrt{nK})$. 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

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

We next show that $\mathbb{E}'[m(a')]$ and $\mathbb{E}[m(a')]$ are close, that is, the number of times a' is selected in environment $\mathcal{E}$ and that in environment $\mathcal{E}'$ are similar (just as in the previous informal argument). Indeed, using $\mathbb{P}$ and $\mathbb{P}'$ to denote the distributions of the observation sequence $\widetilde{\ell}_{1:n}$ in $\mathcal{E}$ and $\mathcal{E}'$ respectively, we have

$$\begin{aligned}\mathbb{E}'[m(a')] - \mathbb{E}[m(a')] &= \sum_{\widetilde{\ell}_{1:n}\in\{0,1\}^n} m(a')\left(\mathbb{P}'(\widetilde{\ell}_{1:n}) - \mathbb{P}(\widetilde{\ell}_{1:n})\right) \le n \sum_{\widetilde{\ell}_{1:n}\in\{0,1\}^n} \left|\mathbb{P}'(\widetilde{\ell}_{1:n}) - \mathbb{P}(\widetilde{\ell}_{1:n})\right| \\ &= n\left\|\mathbb{P}' - \mathbb{P}\right\|_1 \le n\sqrt{2\mathrm{KL}(\mathbb{P} \parallel \mathbb{P}')},\end{aligned}$$

where the last step is by the Pinsker's inequality. To calculate $\mathrm{KL}(\mathbb{P} \parallel \mathbb{P}')$, we apply a handy divergence decomposition lemma (Lemma 3, included after this proof):

$$\begin{aligned}\mathrm{KL}(\mathbb{P} \parallel \mathbb{P}') &= \mathbb{E}[m(a')] \cdot \mathrm{KL}\left(\mathrm{Ber}(1/2) \parallel \mathrm{Ber}(1/2-\epsilon)\right) \\ &= \frac{\mathbb{E}[m(a')]}{2}\left(\ln\frac{1/2}{1/2+\epsilon} + \ln\frac{1/2}{1/2-\epsilon}\right) \\ &= \frac{\mathbb{E}[m(a')]}{2}\ln\left(\frac{1}{1-4\epsilon^2}\right) \\ &\le 8\mathbb{E}[m(a')]\,\epsilon^2,\end{aligned}$$

where in the last step we use the fact $\ln\left(\frac{1}{1-x}\right) \le 4x$ for any $0 \le x \le \frac{1}{2}$. We have thus shown

$$\mathbb{E}'[m(a')] \le \mathbb{E}[m(a')] + 4n\epsilon\sqrt{\mathbb{E}[m(a')]} \le \frac{n}{K} + 4n\epsilon\sqrt{\frac{n}{K}}$$

(recall that a' is selected such that $\mathbb{E}[m(a')] \le n/K$) and thus

$$\mathbb{E}'[\mathrm{Reg}_n] \ge n\left(1 - \frac{1}{K} - 4\epsilon\sqrt{\frac{n}{K}}\right)\epsilon \ge n\left(\frac{1}{2} - 4\epsilon\sqrt{\frac{n}{K}}\right)\epsilon.$$

Setting $\epsilon = \frac{1}{16}\sqrt{\frac{K}{n}}$ (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 3 (Divergence decomposition)**.** *Let $\mathcal{E}$ and $\mathcal{E}'$ be two stochastic MAB environments where for each a, the losses of arm a are i.i.d. samples of $\mathcal{P}_a$ and $\mathcal{P}'_a$ respectively. 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*

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

where $m(a)$ is the total number of times a is selected in $\mathcal{E}$.

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

$$\begin{aligned}\mathrm{KL}(\mathbb{P} \parallel \mathbb{P}') &= \sum_{\widetilde{\ell}_{1:n}} \mathbb{P}(\widetilde{\ell}_{1:n}) \ln\left(\frac{\mathbb{P}(\widetilde{\ell}_{1:n})}{\mathbb{P}'(\widetilde{\ell}_{1:n})}\right) = \sum_{\widetilde{\ell}_{1:n}} \mathbb{P}(\widetilde{\ell}_{1:n}) \ln\left(\frac{\prod_{t=1}^n \mathbb{P}(\widetilde{\ell}_t|\widetilde{\ell}_{1:t-1})}{\prod_{t=1}^n \mathbb{P}'(\widetilde{\ell}_t|\widetilde{\ell}_{1:t-1})}\right) \\ &= \sum_{t=1}^{n} \sum_{\widetilde{\ell}_{1:n}} \mathbb{P}(\widetilde{\ell}_{1:n}) \ln\left(\frac{\mathbb{P}(\widetilde{\ell}_t|\widetilde{\ell}_{1:t-1})}{\mathbb{P}'(\widetilde{\ell}_t|\widetilde{\ell}_{1:t-1})}\right)\end{aligned}$$

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

which completes the proof. □

3 Partial Monitoring

For the final topic of this course, we focus on a problem called *Partial Monitoring*, which allows an extremely general feedback model and greatly generalizes MAB. Indeed, while in MAB the learner observes exactly the loss she suffers, in many other situations the learner only gets to observe information indirectly related to her actual loss. As we will see soon, partial monitoring can be used to capture such more challenging problems.

Specifically, a (finite) partial monitoring problem is defined by a loss matrix $\ell \in [0,1]^{K \times d}$ and a *feedback matrix* $\Phi \in \Sigma^{K \times d}$ where K is the number of actions for the learner, d is the number of outcomes for the environment, and Σ is an arbitrary set of alphabets containing all possible observations for the learner. Both ℓ and Φ are known to the learner. For simplicity, ahead of time, the environment decides n outcomes $z_1, \dots, z_n \in [d] \triangleq \{1, \dots, d\}$ (hence an oblivious environment). Then the learning protocol proceeds in n rounds. For each round $t = 1, \dots, n$, the learner selects an action $a_t \in [K]$, suffers loss $\ell(a_t, z_t)$, and importantly, *only observes* $\Phi(a_t, z_t)$. The goal of the learner is as usual to minimize regret against the best fixed action in hindsight:

$$
\mathrm{Reg}_n = \sum_{t=1}^{n} \ell(a_t, z_t) - \sum_{t=1}^{n} \ell(a^\star, z_t) \quad \text{where } a^\star \in \operatorname*{argmin}_{a \in [K]} \sum_{t=1}^{n} \ell(a, z_t)
$$

The flexibility of having a general feedback matrix Φ allows us to capture many different problems under this framework. Below are a few examples.

Full-information problems. A natural way to encode a learning problem with full information is to set $\Phi(a, z) = z$ for all $a \in [K]$ and $z \in [d]$. That is, no matter what the learner chooses, the actual outcome is observed, which is basically the general online learning protocol we focused on for most of the previous lectures (except that here, for simplicity, both the action space and the outcome space are finite). Note that, however, this is not the only way to encode a full-information problem. In fact, as long as each row of Φ consists of d distinct elements, then it essentially captures the exact same full-information problem, because clearly the learner can still infer the outcome just based on the observation $\Phi(a, z)$.

Bandit problems. Generalizing the well-known multi-armed bandits problem, the term "bandit feedback/information" usually refers to any problems where the learner observes exactly the loss she suffers. These problems can be modeled as partial monitoring by simply setting $\Phi = \ell$. Take MAB as an example. Assuming losses are all binary (which is in fact without loss of generality), we can set $d = 2^K$ and let the columns of ℓ (which is also Φ) represent all the 2^K possible binary vectors in K dimension.

Apple tasting. Apple tasting is one of the simplest problems that are neither full-information nor bandit-information. Imagine a task of classifying a sequence of apples as "good for sale" or "rotten". The loss is 0 if the prediction is correct or 1 otherwise (as in typical binary classification problems). However, only when we predict "rotten" can we actually open the apple and see if it is indeed rotten, otherwise it is sent for sale and we will never know if we predict correctly or not. This can be modeled as partial monitoring by taking

$$\ell = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \qquad \text{and} \qquad \Phi = \begin{pmatrix} \bot & \bot \\ \text{G} & \text{R} \end{pmatrix},$$

where the first (or second) row corresponds to predicting "good for sale" (or "rotten"), and the first (or second) column corresponds to the apple being actually good (or rotten). An observation from the first row of Φ gives no information at all on the actual outcome, while an observation from the second row reveals everything. Note that there are again many other different ways to represent Φ (for example, it can be encoded as $\begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$).

Label efficient learning. Label efficient learning refers to a broad class of problems where querying the true label/outcome is costly and the learner needs to trade-off this cost with information. Consider the following very simple example of classifying emails as spam or not. The spam detector usually does not receive feedback from the users, but in case a very hard instance appears it can choose to ask the user explicitly for answer. Clearly the detector should avoid doing this too often, and to ensure this we can assign a constant loss $c \in (0, 1)$ for each query from the user. Therefore, we can model the problem as partial monitoring by taking

$$\ell = \begin{pmatrix} 0 & 1 \\ 1 & 0 \\ c & c \end{pmatrix} \qquad \text{and} \qquad \Phi = \begin{pmatrix} \bot & \bot \\ \bot & \bot \\ \text{☺} & \text{☹} \end{pmatrix}, \tag{3}$$

where the last row corresponds to querying the user, which is the only action that reveals the true outcome, but on the other hand incurs loss c no matter what the outcome is.

Dynamic pricing. As the last example, consider a vendor trying to dynamically adjust the price of a product with the goal of maximizing revenue. Specifically, each day the vendor first decides a price a_t, say either $1, 2, \ldots$, or K dollars, then a costumer arrives with a secrete acceptable price z_t in mind, also in $[K]$ for simplicity (so $d = K$). The costumer purchases the product if and only if $a_t \le z_t$, in which case the loss of the vendor is $z_t - a_t$, the extra money she would have been able to earn if the product had been priced higher at z_t. Otherwise, no transaction happens and the vendor pays for some constant loss $c > 0$ (for storage fee for example). Importantly, the vendor only observes binary feedback: whether the transaction happens or not, and in particular, she does not know the actual loss she suffers if a transaction happens. To model this as partial monitoring, we can take

$$\ell = \begin{pmatrix} 0 & 1 & 2 & \cdots & K-1 \\ c & 0 & 1 & \cdots & K-2 \\ c & c & 0 & \cdots & K-3 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c & c & c & \cdots & 0 \end{pmatrix} \qquad \text{and} \qquad \Phi = \begin{pmatrix} \checkmark & \checkmark & \checkmark & \cdots & \checkmark \\ \text{✗} & \checkmark & \checkmark & \cdots & \checkmark \\ \text{✗} & \text{✗} & \checkmark & \cdots & \checkmark \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \text{✗} & \text{✗} & \text{✗} & \cdots & \checkmark \end{pmatrix}.$$

4 Classification Theorem

By now you should be convinced that partial monitoring is for sure general enough to capture many problems, but is it too general to be meaningful/solvable? Indeed, clearly there are examples where sublinear regret is impossible, such as

$$\ell = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \qquad \text{and} \qquad \Phi = \begin{pmatrix} \bot & \bot \\ \bot & \bot \end{pmatrix}, \tag{4}$$

because essentially the learner receives no feedback at all and cannot figure out the better action. On the other hand, there are also trivial problems such as

$$\ell = \begin{pmatrix} 0 & 0 \\ 1 & 1 \end{pmatrix} \qquad \text{and} \qquad \Phi = \begin{pmatrix} \bot & \bot \\ \bot & \bot \end{pmatrix}, \tag{5}$$

where even though the learner receives no feedback at all, she can ensure exactly 0 regret by always picking the first action (since it is always better than the second one no matter what the outcome is). Furthermore, we have also discussed that for full/bandit-information problems, by applying Hedge/Exp3 we can achieve $\mathcal{O}(\sqrt{n})$ regret, which is optimal.

So based on these observations we know that there are different kinds of partial monitoring problems with different minimax regret bounds. A somewhat surprising result is that there are exactly four different kinds of such problems — we have seen three of them, with minimax regret 0, $\Theta(\sqrt{n})$, and $\Theta(n)$, and it turns out there is exactly one more class of problems with minimax regret $\Theta(n^{2/3})$. This result, known as the classification theorem, is our focus for the rest of the discussion. To formally introduce it, we need a couple of concepts (examples are deferred to the end of this section).

Cell decomposition First, let's consider when an action can be optimal. Note that the benchmark we want to compare with in the regret definition is

$$\min_{a\in[K]} \sum_{t=1}^{n} \ell(a, z_t) = n \min_{a\in[K]} \langle \ell_a, u \rangle$$

where we use ℓ_a to denote the a-th row of ℓ and $u = \frac{1}{n}\sum_{t=1}^{n} e_{z_t} \in \Delta(d)$ to denote the outcome frequency distribution ($e_1, \ldots, e_d$ are standard basis vectors). The *cell* associated with an action a is then defined as the set of all frequency distributions where a is optimal:

$$C_a = \left\{ u \in \Delta(d) : a \in \operatorname*{argmin}_{a^\star\in[K]} \langle \ell_{a^\star}, u \rangle \right\},$$

which is a polytope since the constraint $a \in \operatorname{argmin}_{a^\star\in[K]} \langle \ell_{a^\star}, u\rangle$ is equivalent to linear constraints $\langle \ell_a - \ell_b, u\rangle \le 0, \forall b \ne a$. All the cells $C_1, \ldots, C_K$ constitute a cell decomposition of $\Delta(d)$. An action a is called

- *dominated*, if $C_a = \emptyset$ (so a is never optimal);
- *degenerate*, if there exists a different action b such that $\emptyset \ne C_a \subsetneq C_b$ (so a is never uniquely optimal);
- *Pareto-optimal*, if it is neither dominated nor degenerate;
- *duplicate*, if there exists a different action b such that $\ell_a = \ell_b$ (or equivalently $C_a = C_b$).

Note that none of the dominated, degenerate, or duplicate actions can be simply ignored, because playing them might provide useful feedback. However, for simplicity, we always assume that there is no degenerate or duplicate actions.[1]

Note that C_a is a $(d-1)$-dimensional polytope if a is Pareto-optimal. Two Pareto-optimal actions a and b are *neighbors* if $C_a \cap C_b$ is of $(d-2)$ dimension. The neighborhood action set of two neighboring actions a and b is generally defined as $N_{ab} = \{k \in [K] : C_a \cap C_b \subseteq C_k\}$. However, under our simplifying assumptions (that there are no degenerate or duplicate actions), N_{ab} is simply $\{a, b\}$.

Signal Matrices The next concepts are related to the feedback matrix Φ (note that all concepts above are only related to the loss matrix ℓ). As we have seen already, there are many equivalent ways to represent the feedback matrix Φ, and the actual symbol of the feedback is unimportant. In fact, all that matters is that after taking a certain action, which outcomes would lead to the same feedback. To this end, we rename the alphabets in Σ as $1, \ldots, |\Sigma|$ (in an arbitrary manner) and let the *signal matrix* $S_a \in \{0,1\}^{|\Sigma|\times d}$ associated with an action a be such that the z-th column is $e_{\Phi(a,z)}$.

Clearly, each column of S_a has exactly one 1. Also, S_a has at most d rows that are not all-zero vectors. If S_a has exactly d such non-zero rows, then picking action a completely reveals the true outcome; otherwise, picking action a might lead to ambiguous feedback. Moreover, note that if the environment chooses an outcome from a distribution $u \in \Delta(d)$, then the distribution of the learner's observation after picking action a can be conveniently written as $S_a u$.

[1] The existence of such actions makes the discussion more involved but does not change the main results.

Finally, for any subset of actions $N \subseteq [K]$, we let $S_N \in \{0,1\}^{(|N||\Sigma|)\times d}$ be the matrix by stacking signal matrices S_a for all $a \in N$.

Observability Now we are ready to define the key concept of observability. A pair of neighboring actions a and b is *globally observable* if $\ell_a - \ell_b \in \text{rowspace}(S_{[K]})$, which is equivalent to either of the following two statements:

- there exists $v_{ab} \in \mathbb{R}^{K|\Sigma|}$ such that $\ell_a - \ell_b = v_{ab}^\top S_{[K]}$;
- there exists a function $v_{ab} : [K] \times \Sigma \to \mathbb{R}$ such that

$$\ell(a,z) - \ell(b,z) = \sum_{k\in[K]} v_{ab}(k, \Phi(k,z)), \quad \forall z \in [d]. \tag{6}$$

Each of these equivalent statements is useful in different ways, but Equation (6) is probably the most intuitive one. Roughly speaking, it says that we can always estimate the loss *difference* between action a and b, no matter what the outcome is. Indeed, suppose that at time t the learner selects an action a_t according to a fully supported distribution $p_t \in \Delta(K)$, then clearly $\frac{v_{ab}(a_t,\Phi(a_t,z_t))}{p_t(a_t)}$ is an unbiased estimator of $\ell(a,z_t) - \ell(b,z_t)$, regardless what z_t is.

Similarly, a pair of neighboring actions a and b is *locally observable* if $\ell_a - \ell_b \in \text{rowspace}(S_{N_{ab}})$ (which is a stronger condition compared to global observability). In other words, Equation (6) holds with $v_{ab}(k,\cdot) = 0$ for all $k \notin N_{ab}$, and thus we can estimate the loss difference between a and b by sampling from a distribution that is *only supported on a and b*.

Finally, a partial monitoring problem is called globally (or locally) observable if every pair of neighboring actions is globally (or locally) observable. Note that if a problem is globally observable, then we also have $\ell_a - \ell_b \in \text{rowspace}(S_{[K]})$ for *any* pair of Pareto-optimal actions a and b (not just neighboring pair, since we can find a sequence of neighboring pairs connecting a and b). We are now ready to state the classification theorem.

Theorem 3 (Classification Theorem)**.** *The minimax regret of a partial monitoring problem G is (ignoring dependence on all parameters but n):*

$$\inf_{\text{learner}} \max_{z_{1:n}} \mathbb{E}\left[\text{Reg}_n\right] = \begin{cases} 0, & \text{if } G \text{ has only one Pareto-optimal action;} \\ \Theta(\sqrt{n}), & \text{else if } G \text{ is locally observable;} \\ \Theta(n^{\frac{2}{3}}), & \text{else if } G \text{ is globally observable;} \\ \Theta(n), & \text{else.} \end{cases}$$

The proof of the classification theorem will be the focus of the next lecture. Here, with the theorem and all the related concepts in mind, we go over each example mentioned in the last section again.

Full-information problems. Since for a full-information problem we can set $\Phi(a,z) = z$ for all $a \in [K]$ and $z \in [d]$, we have that the signal matrix S_a is exactly the identity matrix for any a, and thus any vector in d dimension is in the full-rank row space of S_a. Therefore, regardless of the loss matrix, any full-information problem is locally observable, and thus $\mathcal{O}(\sqrt{n})$ regret is achievable (as we already know since Hedge can be applied).

Bandit problems. Bandit problems with $\ell = \Phi$ are also always locally observable regardless of the loss matrix. To see this, it is more convenient to use Equation (6). Indeed, we can set $v_{ab}(a,\Phi(a,z)) = \Phi(a,z) = \ell(a,z)$, $v_{ab}(b,\Phi(b,z)) = -\Phi(b,z) = -\ell(b,z)$, and $v_{ab}(k,\cdot) = 0$ for all other actions k, so that Equation (6) holds clearly.

Apple tasting. For apple tasting, we have $C_1 = \{u \in \Delta(2) : u_1 \geq u_2\}$ and $C_2 = \{u \in \Delta(2) : u_1 \leq u_2\}$, and both actions are Pareto-optimal and neighbors. Since action 2 reveals all information, the rows of S_2 span $\mathbb{R}^2$ and thus $\ell_1 - \ell_2 \in \text{rowspace}(S_2)$. Therefore, apple tasting is locally observable with $\Theta(\sqrt{n})$ minimax regret.

Label efficient learning. For the simple label efficient learning instance defined in Equation (3), the observability turns out to depend on the value of the query cost c. If $c < 1/2$, then $C_1 =$

$\{u \in \Delta(2) : u_2 \leq c\}$, $C_2 = \{u \in \Delta(2) : u_1 \leq c\}$, and $C_3 = \{u \in \Delta(2) : c \leq \min\{u_1, u_2\}\}$ (try to draw an illustrative figure to help understand this). All actions are Pareto-optimal, and there are two neighboring pairs: actions 1 and 3, and actions 2 and 3. Similarly to apple tasting, since action 3 reveals all information, the rows of S_3 span $\mathbb{R}^2$, and thus $\ell_1 - \ell_3 \in$ rowspace(S_3) and $\ell_2 - \ell_3 \in$ rowspace(S_3), showing that the two neighboring pairs are both locally observable. The minimax regret is therefore $\Theta(\sqrt{n})$ again.

If c is exactly $1/2$, then C_3 has exactly one point (with $u_1 = u_2 = 1/2$) and is contained in C_1 and C_2. So action 3 is degenerate, and this case is beyond the scope of this lecture. Finally, if $c > 1/2$, then $C_3 = \emptyset$ and action 3 is dominated. Actions 1 and 2 are still globally observable (again because $\ell_1 - \ell_2 \in$ rowspace(S_3)), but they are not locally observable since rowspace(S_1) and rowspace(S_2) are both spaces of points with equal coordinates, while $\ell_1 - \ell_2 = (-1, 1)$. This also matches the intuition: playing only actions 1 and 2 is clearly not enough to estimate their loss difference since no information is observed. We conclude that in this case the minimax regret is $\Theta(n^{\frac{2}{3}})$.

Dynamic pricing. For dynamic pricing, it can be verified that regardless of the value of $c > 0$, every pair of actions is neighbor and is globally observable. However, only the pairs $(1, 2), (2, 3), (3, 4), \ldots$ are locally observable. Therefore the minimax regret is $\Theta(n^{\frac{2}{3}})$. In HW4, you will verify this for the special case of $K = 3$.

Finally, we point out that the hopeless problem defined by Equation (4) is clearly not globally observable, so linear regret is unavoidable, and the trivial problem defined by Equation (5) has only one Pareto-optimal action (action 1), so the minimax regret is 0.