Online and Reinforcement Learning (2025)
Home Assignment 1

Davide Marchi 777881

Contents

1 Find an online learning problem from real life

Exercise 5.1 (Find an online learning problem from real life). Find two examples of real life problems that fit into the online learning framework (online, not reinforcement!). For each of the two examples explain what is the set of actions an algorithm can take, what are the losses (or rewards) and what is the range of the losses/rewards, whether the problem is stateless or contextual, and whether the problem is i.i.d. or adversarial, and with full information or bandit feedback.

1.1 Contents Recommendation on Social Media

Deciding in real time which item or advertisement to display to a user.

- **Actions:**
  The algorithm chooses one item from a finite set of options to show to the user.

- **Reward:**
  1 if the user clicks on the suggested content, 0 if the user does not (range [0, 1]).
  In more sophisticated systems, the reward could be based on the time spent by the user on the suggested content.

- **Stateless vs. Contextual:**
  In the simplest implementation, it could be stateless, basing recommendations exclusively on the outcomes of previous interactions. However, in actual implementations, it is contextual, as the algorithm considers the user's profile and related information.

- **Environment (i.i.d. vs. Adversarial):**
  While an idealized model might assume users arrive from an i.i.d. process, real-world user behavior is often adversarial (or non-stationary) as preferences and trends shift over time.

- **Feedback:**
  The feedback is bandit feedback; the algorithm only observes the outcome (click or no click) for the displayed item, not for all items in the set.

1.2 Pricing of Products Over Time

Deciding in real time with what price to sell a product to maximize profit.

- **Set of Actions:**
  Each day, the algorithm can change the price of a product, selecting from a fixed number of price options.

- **Reward:**
  The reward is defined as the profit obtained at the end of the day, which is the revenue from sales minus the cost of production.
  The range of the reward is $(0, +\infty)$ since we assume that none of the prices are lower than the production cost.

- **Stateless vs. Contextual:**
  The problem could be considered stateless if the algorithm only considers the prices and profits of the previous days.

- **Environment (i.i.d. vs. Adversarial):**
  The environment could be considered i.i.d. if the demand for the product remains constant over time. However, in reality, it is adversarial, as demand can change over time, especially for products that are not purchased repeatedly.

- **Feedback Type:**
  The feedback in dynamic pricing is of the bandit type, as the algorithm only observes the reward for the chosen action (applied price).

2 Follow The Leader (FTL) algorithm for i.i.d. full information games

Setting

- We have **two actions**, a^* and a, each with rewards in $[0, 1]$.
- The rewards for each action are *i.i.d. across rounds*, and *fully observed* each round.
- Let $\mu(a)$ be the true expected reward of action a. Without loss of generality, let

$$a^* \;=\; \arg\max_a \; \mu(a).$$

- Define the *gap* of the suboptimal arm by

$$\Delta \;=\; \mu(a^*) - \mu(a) \;>\; 0.$$

- We consider T rounds, indexed by $t = 1, 2, \ldots, T$.

Algorithm: Follow the Leader (FTL)

- **Initialization:** In the very first round, you may pick either (or both) arms arbitrarily.
- **At each round $t \geq 2$:**
  1. For each arm $b \in \{a, a^*\}$, compute its *empirical mean* based on *all* observed rewards of b so far:
  $$\hat{\mu}_{t-1}(b) \;=\; \frac{1}{t-1} \sum_{s=1}^{t-1} X_s(b),$$
  where $X_s(b)$ is the reward observed for arm b at round s.
  2. *Choose* the arm whose empirical mean is larger:
  $$A_t \;=\; \arg\max_{b \in \{a, a^*\}} \; \hat{\mu}_{t-1}(b).$$

Because this is a **full-information** setting, at round t we observe *both* $X_t(a)$ and $X_t(a^*)$, not just the chosen one.

Key Idea: Probability of Picking the Wrong Arm

Define the event

$$E_t \;=\; \big\{\text{“FTL picks the suboptimal arm } a \text{ at round } t\text{”}\big\} \;\iff\; \big\{\,\hat{\mu}_{t-1}(a) \;\geq\; \hat{\mu}_{t-1}(a^*)\big\}.$$

Because $\Delta = \mu(a^*) - \mu(a) > 0$, if $\hat{\mu}_{t-1}(a) \geq \hat{\mu}_{t-1}(a^*)$, then

$$\hat{\mu}_{t-1}(a) \;-\; \hat{\mu}_{t-1}(a^*) \;\;\geq\;\; 0,$$

which can be rewritten as

$$\big[\hat{\mu}_{t-1}(a) - \mu(a)\big] \;-\; \big[\hat{\mu}_{t-1}(a^*) - \mu(a^*)\big] \;\;\geq\;\; \Delta.$$

Let us denote

$$U \;=\; \hat{\mu}_{t-1}(a) - \mu(a), \quad V \;=\; \hat{\mu}_{t-1}(a^*) - \mu(a^*).$$

Then the above condition is $U - V \;\geq\; \Delta$.

Applying Hoeffding's inequality twice. Since U and V are averages of $(t-1)$ i.i.d. $[0,1]$-bounded samples (one set for each arm), Hoeffding's inequality says:

$$\mathbb{P}\big(U \;\geq\; x\big) \;\leq\; \exp\big(-2\,(t-1)\,x^2\big), \quad \mathbb{P}\big(V \;\leq\; -y\big) \;=\; \mathbb{P}\big(-V \;\geq\; y\big) \;\leq\; \exp\big(-2\,(t-1)\,y^2\big).$$

Now, for $U - V \;\geq\; \Delta$ to hold, at least one of the following must happen:

$$U \;\geq\; \tfrac{\Delta}{2} \quad \text{or} \quad -V \;\geq\; \tfrac{\Delta}{2}.$$

Thus, by the union bound,

$$\mathbb{P}\big(U - V \;\geq\; \Delta\big) \;\leq\; \mathbb{P}\Big(U \;\geq\; \tfrac{\Delta}{2}\Big) \;+\; \mathbb{P}\Big(-V \;\geq\; \tfrac{\Delta}{2}\Big).$$

Using Hoeffding on each term separately gives something like

$$\mathbb{P}\Big(U \;\geq\; \tfrac{\Delta}{2}\Big) \;\leq\; \exp\big(-2\,(t-1)\,(\tfrac{\Delta}{2})^2\big) \;=\; \exp\big(-\tfrac{(t-1)\,\Delta^2}{2}\big),$$

and the same for $-V \geq \frac{\Delta}{2}$. Hence

$$\mathbb{P}\big(U - V \;\geq\; \Delta\big) \;\leq\; 2\,\exp\big(-\tfrac{(t-1)\,\Delta^2}{2}\big).$$

We can then absorb the factor of 2 into a constant $c > 0$ and write

$$\mathbb{P}\Big(\hat{\mu}_{t-1}(a) \;\geq\; \hat{\mu}_{t-1}(a^*)\Big) \;=\; \mathbb{P}\big(U - V \;\geq\; \Delta\big) \;\leq\; \exp\big(-\,c\,(t-1)\,\Delta^2\big),$$

for an appropriate $c > 0$.

This is the *probability of confusion* at round t.

Bounding the Regret

Regret definition. The (pseudo-)regret is

$$R_T \;=\; \sum_{t=1}^{T} \Big[\mu(a^*) \;-\; \mu(A_t)\Big].$$

Since $A_t \in \{\, a,\, a^*\}$, the only time we incur regret $\Delta > 0$ is precisely when $A_t = a$. Thus,

$$R_T \;=\; \Delta \sum_{t=1}^{T} \mathbf{1}\{\text{FTL picks } a \text{ at round } t\}.$$

Taking expectation,

$$\mathbb{E}[R_T] \;=\; \Delta \sum_{t=1}^{T} \mathbb{P}\big(A_t = a\big) \;=\; \Delta \sum_{t=1}^{T} \mathbb{P}\Big(\hat{\mu}_{t-1}(a) \;\geq\; \hat{\mu}_{t-1}(a^*)\Big).$$

Using the exponential bound, let us define

$$\delta(t) \;=\; \exp\big(-\,c\,(t-1)\,\Delta^2\big).$$

Hence,

$$\mathbb{E}[R_T] \;\leq\; \Delta \sum_{t=1}^{T} \delta(t).$$

Geometric series. Note that $\sum_{t=1}^{\infty} \delta(t)$ converges, since

$$\sum_{t=1}^{\infty} \exp\big(-\,c\,(t-1)\,\Delta^2\big) \;=\; \frac{1}{1 - e^{-\,c\,\Delta^2}} \;<\; \infty.$$

So there is a constant $C(\Delta)$ such that

$$\sum_{t=1}^{T} \delta(t) \;\leq\; C(\Delta), \quad \text{independently of } T.$$

Multiplying by Δ,

$$\mathbb{E}[R_T] \;\leq\; \Delta\, C(\Delta).$$

Therefore, *the total expected regret* remains **bounded** by a constant with respect to T.

Final Statement of the Bound

Because FTL eventually "locks onto" the better arm and almost never picks the suboptimal one again after some finite time, we conclude

$$R_T \;=\; O(1) \quad \text{as } T \to \infty.$$

An explicit version of the constant bound is

$$\mathbb{E}[\,R_T\,] \;\le\; \Delta\,\frac{1}{1-e^{-\,c\,\Delta^2}} \;=\; \frac{\Delta}{1-e^{-\,c\,\Delta^2}},$$

which is finite for every $\Delta > 0$. If we had more than two arms, we would sum similar terms for each suboptimal gap $\Delta(a)$.

3 Improved Parametrization of UCB1

Below is a figure showing the **empirical pseudo-regret** of two UCB algorithms: "original UCB" (red) vs. "modified UCB" (blue), for three different gap values $\Delta = 0.25, 0.125,$ and 0.0625. Each curve is the average over 20 runs of length $T = 100000$, with shaded ± 1 standard-deviation bands.

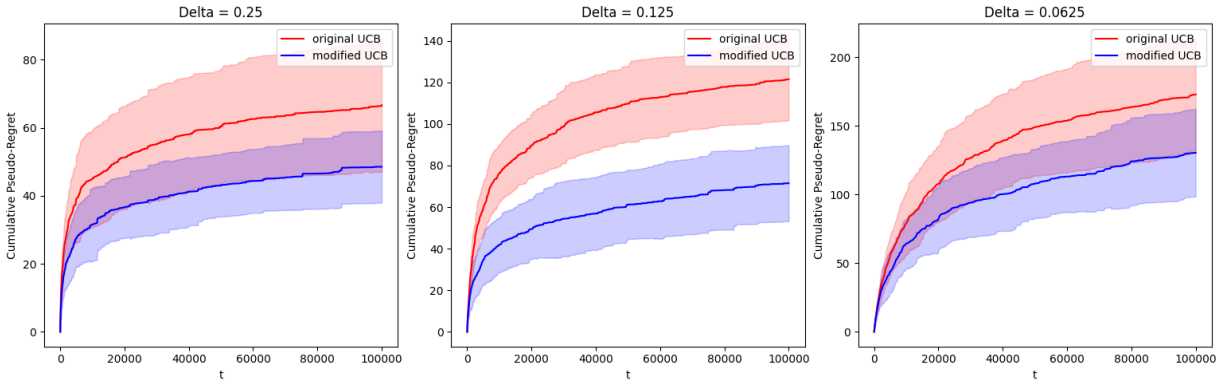

Figure 1: Empirical pseudo-regret over time for different values of Δ.

3.1 Plot Description

- **Horizontal axis**: time $t \in \{1, \ldots, 100000\}$.
- **Vertical axis**: cumulative pseudo-regret $\widehat{R}_t$, i.e. $\sum_{s=1}^{t} \Delta(A_s)$, where $\Delta(A_s)$ is the gap between the chosen arm's mean and the optimal arm's mean at time s.
- **Subplots**: from left to right, we show the same experiment but with $\Delta = 0.25,\ 0.125,$ and 0.0625.

In all three cases, the **blue (modified UCB)** curve remains below the **red (original UCB)** curve, indicating the modified parametrization typically achieves lower pseudo-regret. We also see that for smaller Δ (rightmost plot), overall regret magnitudes are higher.

3.2 1) Which values of Δ lead to higher regret?

We see from the three subplots that:

- $\Delta = 0.25$ (left) yields smaller final regrets, on the order of ≈ 60 for the original UCB and ≈ 40 for the modified UCB by $t = 100000$.
- $\Delta = 0.125$ (middle) has higher final regret (≈ 120 vs. ≈ 60).
- $\Delta = 0.0625$ (right) yields still higher regrets (≈ 170 vs. ≈ 120).

Hence, **the smaller Δ is, the higher the pseudo-regret** tends to be. Intuitively, when the arms' means are closer together, it is harder to distinguish the optimal arm from the suboptimal arm; more exploration is needed before we can reliably exploit.

3.3 2) Relative performance of the two parametrizations

Throughout the plots, the **modified UCB** (blue) consistently achieves lower pseudo-regret than the **original UCB** (red), with the difference becoming more pronounced over time. This matches the theoretical expectation that using a confidence term of $\sqrt{\ln t}$ instead of $\sqrt{\ln t^3}$ results in a tighter exploration bonus, leading to better performance.

The key difference is that the modified parametrization adjusts the confidence radius more efficiently, reducing excessive exploration while still maintaining sufficient uncertainty control. As a result:

- **Modified UCB** adapts the confidence intervals in a way that prioritizes more selective exploration, leading to **lower regret**.
- **Original UCB** uses a slightly larger confidence bound, which results in more cautious exploration and higher cumulative regret over time.

Overall, the modified parametrization demonstrates superior performance across all tested gap values.

4 Example of Policies in RiverSwim

Part 1

We classify the given policies into one of the four policy classes: Π^{SD} (stationary deterministic), Π^{SR} (stationary randomized), Π^{HD} (history-dependent deterministic), and Π^{HR} (history-dependent randomized).

(i) π_a: Swim to the right if the current state is not 1; otherwise swim to the left.

Analysis: This policy depends only on the current state, meaning it is **stationary**. Furthermore, it selects actions deterministically (without randomness).

Conclusion: $\pi_a \in \Pi^{SD}$.

(ii) π_b: If time slot t is an even integer, swim to the right; otherwise, flip a fair coin, then swim to right (resp. left) if the outcome is 'Head' (resp. 'Tail').

Analysis: Since the decision process explicitly depends on t, the policy is **history-dependent**. Additionally, since the agent flips a coin on odd time steps, it involves randomness.

Conclusion: $\pi_b \in \Pi^{HR}$ (history-dependent randomized).

(iii) π_c: At $t = 1$, swim to the left. For $t > 1$, swim to the right if the index of the previous state is odd; otherwise swim to the left. (For $t = 1$, swim to the left.)

Analysis: The decision process depends on both the time step t and the previous state, meaning the policy is **history-dependent**. However, it does not involve randomness.

Conclusion: $\pi_c \in \Pi^{HD}$ (history-dependent deterministic).

(iv) π_d: Flip a fair coin. If the outcome is 'Head' and the current state is either $L-1$ or L, then swim to the right; otherwise swim to the left.

Analysis: The decision depends only on the current state, meaning it is **stationary**. However, since it involves coin flips, it is a **randomized** policy.

Conclusion: $\pi_d \in \Pi^{SR}$ (stationary randomized).

(v) π_e: If it rains, swim to the right; otherwise, swim to the left. (It rains independently of the agent's swimming direction and position.)

Analysis: The decision at each time step depends only on the current state and an external random variable (whether it rains). Since rain is independent of past history, the policy is **stationary** but **randomized**.

Conclusion: $\pi_e \in \Pi^{SR}$ (stationary randomized).

Part 2

Make an arbitrary example of a history-dependent deterministic policy in River-Swim. The policy must not be stationary.

The proposed policy is:

> *At the first visit to state $L-1$, swim to the right. At every subsequent visit to $L-1$, swim to the left. Otherwise, always swim to the right.*

Analysis: This policy is **history-dependent** because the action taken at state $L-1$ depends on whether it is the first visit or a later visit. The decision rule is not simply a function of the current state but requires memory of past visits. Additionally, there is no randomization, so it is **deterministic**.

Conclusion: This policy belongs to Π^{HD} (history-dependent deterministic).

5 Policy Evaluation in RiverSwim

Moved to Home Assignment 2.

6 Robbing Banks

Formulation of the Robber-Police Game as an MDP

(i) State and Action Spaces.
We define each state s as a 4-tuple:

$$s \;=\; \big(r_{\text{agent}},\, c_{\text{agent}},\, r_{\text{police}},\, c_{\text{police}}\big),$$

where

$$r_{\text{agent}},\, r_{\text{police}} \;\in\; \{0,1,2,3,4,5\}, \quad c_{\text{agent}},\, c_{\text{police}} \;\in\; \{0,1,2\}.$$

Therefore, the total number of states is

$$|S| \;=\; 6 \times 3 \;\times\; 6 \times 3 \;=\; 324.$$

We define the initial state as

$$s_{\text{init}} \;=\; (0,\, 0,\, 1,\, 2).$$

The agent has five possible actions at each step:

$$A \;=\; \{\uparrow,\, \downarrow,\, \leftarrow,\, \rightarrow,\, \text{stay}\}.$$

Hence, $|A| = 5$.

(ii) Reward Function.
We define the per-step reward function $R(s,a)$ as follows:

$$R(s,a) \;=\; \begin{cases} 100000, & \text{if the agent is on a bank and the police is not there,} \\ -10000, & \text{if the agent and the police share the same coordinates,} \\ 0, & \text{otherwise.} \end{cases}$$

We then discount the reward by multiplying it by γ^{t-1}, to encourage the agents to chose the best actions from the very beginning.

(iii) Transition Probabilities (with reflecting walls).
We now specify the transition probabilities

$$P\big(s' \mid s, a\big),$$

focusing on the example where the agent is at Bank 1 and the police is at Bank 4. In coordinates, let

$$s \;=\; (\,0,\,0,\,0,\,5).$$

(Here $(0,0)$ denotes Bank 1 in row 0, column 0, and $(0,5)$ denotes Bank 4 in row 0, column 5.)

Wall dynamics. If either the agent or the police attempts to move outside the grid, they *remain in their current cell* (i.e. the move has no effect). For instance:

- If the agent (or police) is at row 0 and an $\uparrow$ action is chosen, they remain in row 0.
- If at column 0 and a $\leftarrow$ action is chosen, they remain in column 0.
- All other moves within the grid follow the usual row or column increment/decrement, bounded between row 0 and row 2, and column 0 and column 5.

Agent's move. From $(r_{\text{agent}},\, c_{\text{agent}}) = (0,\, 0)$, and given the predefined action-space, the next agent position $\big(r'_{\text{agent}},\, c'_{\text{agent}}\big)$ is determined by the followings:

- With probability $\pi(\uparrow \mid s)$: agent tries $\uparrow$; but at row 0, it stays in $(0,0)$.
- With probability $\pi(\leftarrow \mid s)$: agent tries $\leftarrow$; but at column 0, it stays in $(0,0)$.
- With probability $\pi(\rightarrow \mid s)$: agent tries $\rightarrow$; this is valid, leading to $(0,1)$.
- With probability $\pi(\downarrow \mid s)$: agent tries $\downarrow$; this is valid, leading to $(1,0)$.
- With probability $\pi(\text{stay}\mid s)$: agent tries stay; this is valid, leading to $(0,0)$.

Police's move. Meanwhile, since the police is in the same row as the agent but to the right $\big((0,5)\,\text{vs.}\,(0,0)\big)$, the problem statement tells us it moves $\uparrow$, $\downarrow$, or $\leftarrow$ with probability $\frac{1}{3}$ each. Specifically:

- With probability $\frac{1}{3}$: police tries `Up`; but at row 0, it stays in $(0,5)$.
- With probability $\frac{1}{3}$: police tries `Down`; this is valid, leading to $(1,5)$.
- With probability $\frac{1}{3}$: police tries `Left`; this is valid, leading to $(0,4)$.

Combining to get next-state probabilities. Let $\big(r'_{\text{agent}}, c'_{\text{agent}}\big)$ be the agent's new position after action a, and $\big(r'_{\text{police}}, c'_{\text{police}}\big)$ be the result of the police's stochastic move. In general, the full next state is

$$s' \;=\; \big(r'_{\text{agent}},\, c'_{\text{agent}},\, r'_{\text{police}},\, c'_{\text{police}}\big).$$

Because the agent's sequence of actions depends on a policy π (which we do not yet know), we keep $\pi(a \mid s)$ for the probability that action a is chosen in state s.

What we end up with is:

$$\begin{aligned}
P(0,0,?,?) &= \pi(\uparrow|\ s) + \pi(\leftarrow|\ s) + \pi(\text{stay} \mid s) \\
P(0,1,?,?) &= \pi(\rightarrow|\ s) \\
P(1,0,?,?) &= \pi(\downarrow|\ s) \\
P(?,?,0,5) &= \tfrac{1}{3} \\
P(?,?,1,5) &= \tfrac{1}{3} \\
P(?,?,0,4) &= \tfrac{1}{3}
\end{aligned}$$