

## Chapter 9

# Exploration vs Exploitation

*In this chapter we discuss in detail the exploration-exploitation tradeoff, briefly alluded to in the previous chapter. We introduce the multi-armed bandit problem as the simplest instantiation of this tradeoff, and build algorithmic approaches to this class of problems greatly inspired by the problem of sequential prediction. We then extend our discussion of exploration vs. exploitation to the reinforcement learning setting, discussing sample complexity in this class of problems.*

### 9.1 Sequential prediction with expert advice

Before formally discussing the problem of exploration vs. exploitation, we visit simple prediction problems—and the algorithmic machinery used to solve it, as these lay the foundational ideas throughout this chapter.

We start with a motivational example.

---


























**Example 9.1** A broadcasting company wishes to develop a computer system to forecast the weather. Each day the system is expected to predict whether the following day will be “Sunny” or “Rainy”. To build its prediction, the system can access the forecasts from 5 different sources: B-Weather, G-Weather, Y-Weather, A-Weather, and W-Weather. It is known that one of the 5 sources is able to make flawless forecasts, although which source it is is unknown.

The system follows the following simple algorithm:

- Each day, the system forecasts the weather predicted by the majority of the available sources;
- Upon observing the weather in the following day, the system removes from the set of available sources those that made a mistake;

Then, at each day, one of two possible outcomes takes place:

**Table 9.1** Forecasts by the different sources on the first 5 days of work of the new weatherman.

	BW	GW	YW	AW	WW
Day 1					
Day 2					
Day 3					
Day 4					
Day 5					

- The system makes the correct forecast;
- Half or more of the sources still available made the wrong forecast and are, therefore, eliminated from the set of available sources.

It follows that the number of mistakes made by the system is, at most,  $\log_2(5) < 3$ .

To illustrate this situation, the 5 initial forecasts from the 5 sources are summarized in Table 9.1.

On day 1, and without any prior information that would lead him to trust one source more than the others, the system forecasts “Sunny”. The day turns out to be sunny so, in light of the wrong forecast of YW, it is eliminated from consideration.

On day 2, the system forecasts a rainy day, but the day again turns out to be sunny. At this point, BW, AW and WW are eliminated and the system successfully identified GW as the reliable source, following its forecasts thereafter (“Rainy”, “Rainy” and “Sunny”).

Let us formalize the prediction problem illustrated in the example above. An agent must select, at each time step  $t \in \mathbb{N}$ , an action  $a_t \in \mathcal{A}$ , where  $\mathcal{A}$  is a finite set of alternatives. For now we consider the simplest case where  $|\mathcal{A}| = 2$  (in the example,  $\mathcal{A} = \{\text{“Sunny”}, \text{“Rainy”}\}$ ). To aid in the selection of the action, the agent has available a set  $\Pi$  of *predictors*, where a predictor corresponds to a mapping  $\pi : \mathcal{H} \rightarrow \mathcal{A}$  and  $\mathcal{H}$  is the set of all finite-length histories.<sup>1</sup> Then, given a finite

<sup>1</sup>In the present context, the history  $h_t$  is allowed to include arbitrary (past) information. For instance, in the weather example, the predictors take into consideration all relevant meteorological and atmospheric information, weather models, etc.

history  $h_t \in \mathcal{H}$ ,  $\pi(h_t)$  corresponds to the action predicted by  $\pi$  at time step  $t$  after observing  $h_t$ .

In practice, each predictor  $\pi \in \Pi$  corresponds to a *policy*, and the role of the agent is to determine which policies are more adequate for the task at hand. In the example, the policies correspond to the 5 information sources. We henceforth refer to the elements of  $\Pi$  simply as *policies*.

At each time step  $t$ , the agent selects an action

$$a_t = \operatorname{argmax}_{a \in \mathcal{A}} \sum_{\pi \in \Pi} \mathbb{I}[\pi(t) = a] \pi(h_t) w_{t-1}(\pi),$$

where  $w_t(\pi) = 0$  if policy  $\pi$  has made a mistake up to time step  $t$ , and 1 otherwise. The agent then incurs a cost  $c_t(a_t)$ , where  $c_t$  is an unknown cost function, with  $c_t : \mathcal{A} \rightarrow \{0, 1\}$ . A cost  $c_t(a_t) = 0$  corresponds to a “correct” action, while a cost  $c_t(a_t)$  corresponds to a “mistake”. The weight associated with each policy  $\pi \in \Pi$  is then updated as

$$w_t = w_{t-1}(1 - c_t(\pi(h_t))).$$

The algorithm in Example 9.1 is known as the *halving algorithm* and rests on the assumption that there is a policy  $\pi^* \in \Pi$  such that  $c_t(\pi^*(h_t)) = 0$  for all  $t \in \mathbb{N}$  (i.e., a policy that makes no “mistakes”). The result is that, at each time step  $t$ , either  $c_t(a_t) = 0$  or  $W_t < \frac{W_{t-1}}{2}$ , where  $W_t$  is the number of policies with positive weight at time step  $t$ , i.e.,

$$W_t = \sum_{\pi \in \Pi} w_t(\pi). \quad (9.1)$$

In other words, either the agent selects a correct action, or more than half of the policies with positive weight made a mistake and, therefore, the number of such policies is halved when the weights are updated. The consequence is that, as seen in the example, an agent following the halving algorithm will make, at most,  $\log_2 |\Pi|$  mistakes.

The halving algorithm is, perhaps, among the simplest sequential prediction problems. It relies on very stringent assumptions—namely, that there is a policy  $\pi^* \in \Pi$  that makes no “mistakes”. In the continuation, we discuss a more general setting in which we alleviate such assumption.

### 9.1.1 The weighted majority algorithm

Consider an agent engaged in a sequential game with “Nature” where, at each round  $t$ ,

- The agent selects an action from the set of actions  $\mathcal{A} = \{0, 1\}$ . We write  $a_t$  to denote the action of the agent at time step  $t$ ;
- “Nature” selects a cost function  $c_t : \mathcal{A} \rightarrow \{0, 1\}$ ;
- Simultaneously, “Nature” reveals the cost function  $c_t$  and the agent executes the action  $a_t$ ;

- The agent incurs a cost  $c_t(a_t)$ .

As in Example 9.1, the agent has available a finite set  $\Pi$  of policies. Also, we may again interpret that if  $c_t(a_t) = 0$ , then action  $a_t$  is considered “correct”, while if  $c_t(a_t) = 1$  then  $a_t$  is considered a “mistake”. However, unlike Example 9.1, we make no additional assumptions on  $c_t$ —it may actually be selected in an adversarial manner. For this reason, it is no longer possible to directly use the halving algorithm: since no policy is always correct, the weight update rule may lead to a situation where  $w(\pi) = 0$  for all  $\pi \in \Pi$ .

Instead, we modify the update of the weights  $w(\pi)$  to handle the fact that no policy is always correct to get

$$w_t(\pi) = w_{t-1}(\pi)(1 - \beta c_t(\pi(h_t))),$$

for some constant  $\beta \in [0, 1)$ . Then, at each step  $t$ , the agent again selects the action

$$a_t = \operatorname{argmax}_{a \in \mathcal{A}} \sum_{\pi \in \Pi} \mathbb{I}[\pi(t) = a] \pi(h_t) w_{t-1}(\pi).$$

The resulting algorithm is known as the *weighted majority algorithm*. The weight  $w(\pi)$  can be interpreted as the “confidence” of the agent in policy  $\pi$ : a weight closer to 1 indicates a policy that has made few “mistakes”, while a weight close to 0 indicates a policy that has made many “mistakes”. The goal of the agent is to minimize the total incurred cost or, equivalently, minimize the number of “mistakes”. We have the following result.

**Proposition 9.1.** *Given a finite set of policies,  $\Pi$ , let  $m$  denote the number of mistakes incurred by the best policy in  $\Pi$  up to time step  $T > 0$ . Then, the number of mistakes  $M_T$  incurred by an agent following the weighted majority algorithm verifies*

$$M_T \leq \frac{2}{\beta} \log |\Pi| + 2(1 + \beta)m. \quad (9.2)$$

The method of proof for the bound in Proposition 9.1 is illustrative of how such bounds can be established and is, as such, provided in the continuation.

*Proof of Proposition 9.1.* Let  $w_t(\pi)$  denote the weight associated with policy  $\pi$  after time step  $t$ , and let

$$W_t = \sum_{\pi \in \Pi} w_t(\pi).$$

Furthermore, let  $M_t$  denote the number of mistakes incurred by the agent up to time step  $t$ , i.e.,

$$M_t = \sum_{\tau=1}^t c_\tau(a_\tau).$$

At any time step  $t + 1$ , either  $c_{t+1}(a_{t+1}) = 0$  or  $c_{t+1}(a_{t+1}) = 1$ . In the former case,  $M_{t+1} = M_t$  and the action  $a_t$  does not add to the total number of mistakes incurred by the agent.

Let us then consider that  $c_{t+1}(a_{t+1}) = 1$ , and let  $\varepsilon_t$  denote the total weight of all policies  $\pi \in \Pi$  such that  $\pi(h_{t+1}) = a_{t+1}$ , and  $\gamma_t$  the total weight of all policies such that  $\pi(h_{t+1}) \neq a_{t+1}$  before the weights are updated. It follows that  $\varepsilon_t \geq \gamma_t$  and

$$W_t = \varepsilon_t + \gamma_t \leq 2\varepsilon_t. \quad (9.3)$$

After the weight update, we have that

$$W_{t+1} = (1 - \beta)\varepsilon_t + \gamma_t = W_t - \beta\varepsilon_t \leq W_t \left(1 - \frac{\beta}{2}\right),$$

where the last inequality follows from (9.3). Unfolding the recursion and using the fact that  $W_0 = |\Pi|$ , we get that

$$W_{t+1} \leq |\Pi| \left(1 - \frac{\beta}{2}\right)^{M_{t+1}}.$$

On the other hand, let  $\pi^*$  denote the best policy in  $\Pi$  up to time step  $T$ , i.e.,

$$\pi^* = \operatorname{argmax}_{\pi \in \Pi} \{w_T(\pi)\},$$

and let  $m$  denote the number of mistakes incurred by  $\pi^*$  up to  $T$ . Then,

$$w_T(\pi^*) = (1 - \beta)^m \leq W_T \leq |\Pi| \left(1 - \frac{\beta}{2}\right)^{M_T}.$$

Applying the logarithm on both sides and using (A.13), we get

$$m \log_2(1 - \beta) \leq \log |\Pi| + M_T \log \left(1 - \frac{\beta}{2}\right) \leq \log |\Pi| - M_T \frac{\beta}{2}.$$

Finally, solving for  $M_T$  and using (A.14),

$$M_T \leq \frac{2}{\beta} \log |\Pi| + 2(1 + \beta)m.$$

□

### 9.1.2 The randomized weighted majority

One problem with the weighted majority algorithm lies in the fact that the action selection rule is *deterministic*. Therefore, if the cost function  $c_t$  is, indeed, selected in an adversarial way, it is possible for “Nature” to exploit the agent. In fact, comparing the performance of the agent with that of the best policy, we get

$$M_T - m \leq \frac{2}{\beta} \log |\Pi| + (1 + 2\beta)m. \quad (9.4)$$

Assuming that the best policy fails, in average, a percentage  $\rho_0$  of time—i.e., letting  $m = \rho_0 T$  for some  $\rho_0 > 0$ —and setting  $\beta = \sqrt{\log |\Pi| / m}$ ,<sup>2</sup> we get

$$M_T - m \leq \rho_0 T + \rho_1 \sqrt{\log |\Pi| T}.$$

<sup>2</sup>The value for  $\beta$  is obtained by minimizing the right-hand side of (9.4).

---

**Algorithm 9.1** Randomized weighted majority.

---

**Require:** Set of candidate policies,  $\Pi$ .

**Require:** Update ratio,  $\beta \in [0, \frac{1}{2}]$ .

```

1: for all  $\pi \in \Pi$  do
2:   Initialize  $w(\pi) \leftarrow 1$ 
3: end for
4: for  $t = 1, \dots, T$  do
5:   Randomly select a policy  $\pi_t \in \Pi$  according to the distribution

```

$$p_t(\pi) = \frac{w(\pi)}{\sum_{\pi \in \Pi} w(\pi)}. \quad (9.6)$$

```

6:   Observe costs  $c_t(a), a \in \mathcal{A}$ 
7:   for all  $\pi \in \Pi$  do
8:     Update weight  $w(\pi)$  as

```

$$w(\pi) \leftarrow w(\pi)(1 - \beta c_t(\pi(h_t))). \quad (9.7)$$

```

9:   end for
10: end for

```

---

for some constant  $\rho_1 > 0$ . Therefore, the difference in performance between the agent and the best policy grows linearly with  $T$ .

By using randomization, however, it is possible to avoid being exploited by an adversarial selection of  $c_t$ . Instead of following the majority vote, the agent randomly selects a policy  $\pi \in \Pi$  at each time step  $t$ . Specifically, at each time step  $t$  the agent selects a policy  $\pi_t$  according to the distribution

$$p_t(\pi) \stackrel{\text{def}}{=} \mathbb{P}[\pi_t = \pi] = \frac{w_{t-1}(\pi)}{W_{t-1}}, \quad (9.5)$$

where  $W_t$  is defined in (9.1). The weights are now used to ensure that those policies that are correct more often are selected more frequently. The resulting approach is summarized in Algorithm 9.1, and is known as the *randomized weighted majority* algorithm.

To analyze the performance of Algorithm 9.1, we again compare the performance of the agent with that of the best policy in hindsight. We define the *expected regret* of the agent as the difference

$$R_T = \sum_{t=1}^T \mathbb{E}_{p_t} [c_t(\pi_t(h_t))] - \min_{\pi \in \Pi} \sum_{t=1}^T c_t(\pi(h_t)),$$

where the expectation is taken with respect to the randomization in the policy selection. The expected regret compares the expected number of correct actions of Algorithm 9.1 with that of the best policy in hindsight. Once again, letting  $m$  denote the number of mistakes of the such policy, we have the following result.

**Proposition 9.2.** *After  $T$  time steps, the expected regret of Algorithm 9.1 verifies*

$$R_T \leq \frac{1}{\beta} \log |\Pi| + m\beta. \quad (9.8)$$

If, moreover,  $\beta = \sqrt{\frac{\log |\Pi|}{m}}$ ,

$$R_T \leq 2\sqrt{m \log |\Pi|}.$$

*Proof.* The proof of Proposition 9.2 is mostly the same as that of Proposition 9.1 and is provided in Section 9.3.  $\square$

Unlike the bound in Proposition 9.1, by a proper selection of  $\beta$  it is possible to ensure that the regret is *sub-linear* in  $T$ . In fact, if  $m$  grows linearly with  $T$ , we get that

$$R_T \leq \rho \sqrt{T \log |\Pi|},$$

for some positive constant  $\rho$ . Additionally, the average expected regret per time step thus becomes

$$\frac{R_T}{T} \leq \rho \sqrt{\frac{\log |\Pi|}{T}},$$

and we have that

$$\lim_{T \rightarrow \infty} \frac{R_T}{T} = 0.$$

An algorithm for which the expected average reward per time step is asymptotically 0 is called a *no-regret algorithm*.

We note, however, that the bound provided in Proposition 9.2 concerns the worst case performance. In many practical situations, the environment is not adversarial, the performance of the non-randomized weighted majority algorithm may actually be superior—even though the worst-case regret bounds are worse.

### 9.1.3 The exponentially weighted averager

So far we considered only the situation in which the cost function takes values in  $\{0, 1\}$ . However, in many practical situations it may be convenient to consider richer cost functions that penalize different actions in  $\mathcal{A}$  differently.

We thus consider the problem of an agent engaged in a sequential game with “Nature” where, at each round  $t$ ,

- The agent selects an action from the (finite) set of actions  $\mathcal{A}$ . We write  $a_t$  to denote the action of the agent at time step  $t$ ;
- “Nature” selects a cost function  $c_t : \mathcal{A} \rightarrow [0, 1]$ ;
- Simultaneously, “Nature” reveals the cost function  $c_t$  and the agent executes the action  $a_t$ ;

- The agent incurs a cost  $c_t(a_t)$ .

We again assume that the agent has available a finite set  $\Pi$  of policies. Each policy in  $\Pi$  is now a mapping  $\pi : \mathcal{H} \rightarrow \Delta(\mathcal{A})$ , where  $\Delta(\mathcal{A})$  denotes the set of possible distributions over  $\mathcal{A}$ . In other words, we allow for the possibility of stochastic policies in  $\Pi$ . We write  $\pi(a | h)$  to denote the probability of action  $a$  under  $\pi$  when history  $h$  is observed.

Each policy  $\pi \in \Pi$  has an associated weight  $w(\pi)$  that can be interpreted as the “confidence” that the agent has on policy  $\pi$ . However, since the cost can now take any value between 0 and 1, it is no longer convenient to refer to an action as being “correct” or a “mistake”.

We adopt the same approach used in randomized weighted majority to select the actions: at each step  $t$ , the agent selects a random action  $a_t$ , according to the distribution

$$p_t(a) \stackrel{\text{def}}{=} \mathbb{P}[a_t = a] = \sum_{\pi \in \Pi} \frac{w_{t-1}(\pi)}{W_{t-1}} \pi(a | h_t). \quad (9.9)$$

For deterministic policies, (9.9) reduces to (9.5).

Then, upon observing the cost function  $c_t$ , the weights  $w(\pi)$  are updated according to the rule

$$w_t(\pi) = w_{t-1}(\pi) e^{-\eta c_{\pi,t}},$$

where  $\eta$  is a positive constant and

$$c_{\pi,t} = \mathbb{E}_{a \sim \pi(h_t)} [c_t(a)] \stackrel{\text{def}}{=} \sum_{a \in \mathcal{A}} c_t(a) \pi(a | h_t).$$

The multiplicative update is now dependent on the cost incurred by  $\pi$ : policies incurring larger costs will have their weight more attenuated. The resulting algorithm is known as the *exponentially weighted averager* (EWA) and is summarized in Algorithm 9.2.

To assess the performance of the EWA, we bound the corresponding expected regret. The following result is an extension of Proposition 9.2 to the exponentially weighted averager.

**Proposition 9.3.** *After  $T$  time steps, the expected regret of Algorithm 9.2 verifies*

$$R_T \leq \frac{\log |\Pi|}{\eta} + \frac{\eta}{8} T.$$

If, moreover,  $\eta = \sqrt{\frac{8 \log |\Pi|}{T}}$ ,

$$R_T \leq \sqrt{\frac{T}{2} \log |\Pi|}.$$

*Proof.* See Section 9.3. □



---

**Algorithm 9.2** Exponentially weighted averager (EWA).

---

**Require:** Set of candidate policies,  $\Pi$ .**Require:** Precision parameter,  $\eta > 0$ .

- 1: **for all**  $\pi \in \Pi$  **do**
- 2:     Initialize  $w(\pi) \leftarrow 1$
- 3: **end for**
- 4: **for**  $t = 1, \dots, T$  **do**
- 5:     Randomly select an action  $a_t \in \mathcal{A}$  according to the distribution

$$p_t(a) = \frac{\sum_{\pi \in \Pi} w(\pi) \pi(a \mid h_t)}{\sum_{\pi \in \Pi} w(\pi)}.$$

- 6:     Observe costs  $c_t(a), a \in \mathcal{A}$
- 7:     **for all**  $\pi \in \Pi$  **do**
- 8:         Update weights  $w(\pi)$  as

$$w(\pi) \leftarrow w(\pi) e^{-\eta c_{\pi,t}}. \quad (9.10)$$

- 9:     **end for**

- 10: **end for**
- 

The regret bound obtained for the EWA is similar to that obtained for the randomized weighted majority algorithm. It also features a sub-linear dependence on  $T$  and a logarithmic dependence on the number of experts,  $|\Pi|$ . Additionally, the EWA is also a no-regret algorithm, since

$$\lim_{T \rightarrow \infty} \frac{R_T}{T} \leq \lim_{T \rightarrow \infty} \sqrt{\frac{\log |\Pi|}{2T}} = 0.$$

#### 9.1.4 Multi-armed bandits with expert advice

In the sequential prediction problems considered so far, the agent is able to observe, at each time step  $t$ , the full cost function  $c_t$  selected by “Nature”. We now consider the situation where the agent is only able to observe the cost associated with its action, i.e.,  $c_t(a_t)$ . Such setting is commonly referred as a *multi-armed bandit problem*.

##### Multi-armed bandit problem

A *multi-armed bandit problem* is a sequential game between an agent and “Nature” where, at each round  $t$ ,

- The agent selects an action from the (finite) set of actions  $\mathcal{A}$ . We write  $a_t$  to denote the action of the agent at time step  $t$ ;

- “Nature” selects a cost function  $c_t : \mathcal{A} \rightarrow [0, 1]$ ;
- The agent executes the action  $a_t$  and incurs a cost  $c_t(a_t)$ .

The distinctive elements of a multi-armed bandit problem are, therefore, the sequential nature of the decision problem and the lack of information regarding the cost of the actions not selected by the agent at each time step  $t$ .

In the present section, we address the multi-armed bandit problem as a generalization of the prediction problems considered so far. In particular, we make no assumptions regarding the process by which the cost function  $c_t$  is selected. Additionally, we assume that the agent has available a finite set  $\Pi$  of policies (possibly stochastic), where each policy  $\pi \in \Pi$  has an associated weight  $w(\pi)$ . In such setting the multi-armed bandit problem is very similar to the sequential prediction problems discussed in the previous section. Unfortunately, as will soon become apparent, the EWA algorithm cannot be directly applied to the bandit problem.

EWA updates all the weights  $w(\pi), \pi \in \Pi$  according to the cost of policy  $\pi$  at time step  $t$ ,  $c_{\pi,t}$ . In other words,

$$w_t(\pi) = w_{t-1}(\pi) \exp \{-\eta c_{\pi,t}\} = w_{t-1} \exp \left\{ -\eta \sum_{a \in \mathcal{A}} c_t(a) \pi(a \mid h_t) \right\}.$$

The difficulty in the multi-armed bandit setting is now apparent: since the agent can only observe the cost associated with the action it executed, the term  $c_{\pi,t}$  cannot be directly computed to perform the update.

However, it is possible to use the cost information available to the agent to build an unbiased estimate of  $c_t(a), a \in \mathcal{A}$ . Using simple algebraic manipulations, we have

$$\begin{aligned} c_t(a) &= \sum_{a' \in \mathcal{A}} c_t(a') \mathbb{I}[a' = a] \\ &= \sum_{a' \in \mathcal{A}} \frac{c_t(a')}{p_t(a')} \mathbb{I}[a' = a] p_t(a') \\ &= \mathbb{E}_{p_t} \left[ \frac{c_t(a)}{p_t(a)} \mathbb{I}[a = a] \right]. \end{aligned}$$

This means that the random variable  $\hat{c}_t(a)$ , defined as

$$\hat{c}_t(a) = \frac{c_t(a_t)}{p_t(a_t)} \mathbb{I}[a = a_t], \quad (9.11)$$

is an unbiased estimator of  $c_t(a)$ , and we can use it to perform the desired weight update. The use of the estimator in (9.11) with EWA leads to the algorithm known as *exponential-weighting for exploration and exploitation with experts*, or EXP4. We summarize EXP4 in Algorithm 9.3.

To assess the performance of EXP4, we provide an upper bound on the corresponding expected regret, summarized in the following result.

---

**Algorithm 9.3** The EXP4 algorithm.

---

**Require:** Set of candidate policies,  $\Pi$ .**Require:** Precision parameter,  $\eta > 0$ .

- 1: **for all**  $\pi \in \Pi$  **do**
- 2:     Initialize  $w(\pi) \leftarrow 1$
- 3: **end for**
- 4: **for**  $t = 1, \dots, T$  **do**
- 5:     Randomly select an action  $a_t \in \mathcal{A}$  according to distribution

$$p_t(a) = \frac{\sum_{\pi \in \Pi} w(\pi) \pi(a \mid h_t)}{\sum_{\pi \in \Pi} w(\pi)}.$$

- 6:     Observe cost  $c_t(a_t)$
- 7:     Let

$$\hat{c}_t(a) = \frac{c_t(a)}{p_t(a)} \mathbb{I}[a = a_t]$$

- 8:     **for all**  $\pi \in \Pi$  **do**
- 9:         Update weights  $w(\pi)$  as

$$w(\pi) \leftarrow w(\pi) e^{-\eta \hat{c}_{\pi,t}},$$

where

$$\hat{c}_{\pi,t} = \sum_{a \in \mathcal{A}} \hat{c}_t(a) \pi(a \mid h_t).$$

- 10:     **end for**
  - 11: **end for**
- 

**Proposition 9.4.** *After  $T$  time steps, the expected regret of Algorithm 9.3 verifies*

$$R_T \leq \frac{\log |\Pi|}{\eta} + \frac{\eta}{2} T |\mathcal{A}|.$$

If, moreover,  $\eta = \sqrt{\frac{2 \log |\Pi|}{T |\mathcal{A}|}}$ ,

$$R_T \leq \sqrt{2T |\mathcal{A}| \log |\Pi|}.$$

*Proof.* See Section 9.3. □

It is educative to compare the regret bounds for EXP4 and EWA. The two are quite similar, the only difference being the dependence of EXP4 on  $\mathcal{A}$ . This is to be expected, since EXP4 receives, at each time step,  $1/|\mathcal{A}|$  the amount of (cost) information that EWA receives. In any case, the regret of EXP4 is also sub-linear on  $T$ , implying that EXP4 is also a no-regret algorithm.

◇

The algorithms considered in this section—from the halving algorithm to EXP4—are usually introduced in the context of *prediction from expert advice*. In such context, the agent is expected to predict some quantity of interest (the action) for which it can resort to the predictions provided by a number of experts (the policies in  $\Pi$ ).

In the case where the agent is expected to select an action but no experts/policies are provided, the algorithms can still be applied by considering the set of constant policies,  $\Pi = \{\pi_a, a \in \mathcal{A}\}$  where, for each  $a \in \mathcal{A}$ ,  $\pi_a(h_t) = a$  for all histories  $h_t \in \mathcal{H}$ . In other words, each policy in  $\Pi$  corresponds to an action of the agent, and everything else in the algorithms remains unchanged. In the next section, we revisit the multi-armed bandit problem when the agent has no “experts” available.

## 9.2 Multi-armed bandits

We now explore in greater detail the multi-armed bandit problem introduced in the previous section. Recall that a multi-armed bandit problem is a sequential game between an agent and “Nature” where, at each round  $t$ ,

- The agent selects an action from the (finite) set of actions  $\mathcal{A}$ . We write  $a_t$  to denote the action of the agent at time step  $t$ ;
- “Nature” selects a cost function  $c_t : \mathcal{A} \rightarrow [0, 1]$ ;
- The agent executes the action  $a_t$  and incurs a cost  $c_t(a_t)$ .

Section 9.1 introduced the multi-armed bandit problem as a more general version of the sequential prediction problem with expert advice. We now drop the assumption that the agent has available a set of “experts” that are used to make its prediction.

We consider two different classes of bandit problems. The first class, known as *adversarial multi-armed bandit problems*, is similar to the problems in Section 9.1; we make no assumptions on the process by which the costs  $c_t$  are selected. The second class, known as *stochastic multi-armed bandit problems*, deals with the particular case in which the costs  $c_t$  follow some underlying distribution.

### 9.2.1 Adversarial multi-armed bandits

As mentioned in Section 9.1, a bandit problem in which the agent is expected to select an action while having no “experts” available can be converted into an equivalent problem with a set of “experts” corresponding to the constant policies. In such equivalent problem, the algorithms discussed in Section 9.1 can be applied with no modification, and the corresponding performance guarantees still hold.

Bearing such equivalence in mind, we associate with each action  $a \in \mathcal{A}$  a weight  $w(a)$ . The weights translate how “reliable” the different actions are, and the agent must experiment the different actions to update the corresponding weights (explore). However, as the agent becomes more knowledgeable regarding the success of the different actions, it should take advantage of such information and select more often the potentially better actions (exploit). Solving the multi-armed bandit

---

**Algorithm 9.4** The EXP3 algorithm.

---

**Require:** Set of candidate policies,  $\Pi$ .**Require:** Precision parameter,  $\eta > 0$ .

- 1: **for all**  $\pi \in \Pi$  **do**
- 2:     Initialize  $w(\pi) \leftarrow 1$
- 3: **end for**
- 4: **for**  $t = 1, \dots, T$  **do**
- 5:     Randomly select an action  $a_t \in \mathcal{A}$  according to distribution

$$p_t(a) = \frac{w(a)}{\sum_{a' \in \mathcal{A}} w(a')}.$$

- 6:     Observe cost  $c_t(a_t)$
- 7:     Let

$$\hat{c}_t(a) = \frac{c_t(a)}{p_t(a)} \mathbb{I}[a = a_t]$$

- 8:     **for all**  $a \in \mathcal{A}$  **do**
- 9:         Update weights  $w(a)$  as

$$w(a) \leftarrow w(a)e^{-\eta \hat{c}_t(a)}.$$

- 10:     **end for**
  - 11: **end for**
- 

problem thus consists of finding an adequate trade-off between the two conflicting courses of action—explore and exploit.

Translating the EXP4 algorithm to the present setting, at each time step  $t$  the agent selects a random action  $a_t$  according to the distribution

$$p_t(a) = \frac{w_{t-1}(a)}{W_{t-1}}$$

where, for all  $t$ ,

$$W_t = \sum_{a \in \mathcal{A}} w_t(a).$$

Upon observing the cost  $c_t(a_t)$ , the agent computes the random variable  $\hat{c}_t(a)$ , defined as

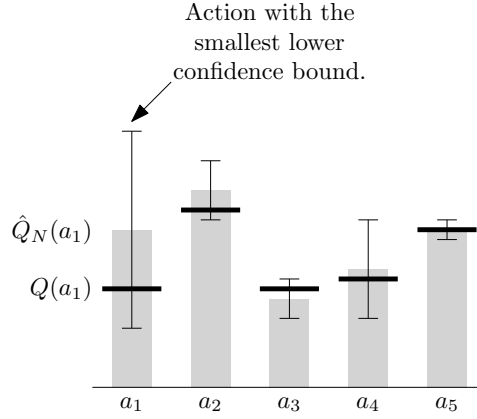
$$\hat{c}_t(a) = \frac{c_t(a_t)}{p_t(a_t)} \mathbb{I}[a = a_t],$$

and performs the update

$$w(a) \leftarrow w(a)e^{-\eta \hat{c}_t(a)}.$$

The algorithm is known as *exponential-weighting for exploration and exploitation*, or EXP3, and is summarized in Algorithm 9.4.

The performance of EXP3 is assessed in the following result.



**Figure 9.1** Optimism in the face of uncertainty: the agent selects not the action with the smallest estimated cost (action  $a_3$ ), but the action with the smallest *lower confidence bound* (action  $a_1$ ).

**Proposition 9.5.** *After  $T$  time steps, the expected regret of Algorithm 9.4 verifies*

$$R_T \leq \frac{\log |\mathcal{A}|}{\eta} + \frac{\eta}{2} T |\mathcal{A}|.$$

If, moreover,  $\eta = \sqrt{\frac{2 \log |\mathcal{A}|}{T |\mathcal{A}|}}$ ,

$$R_T \leq \sqrt{2T |\mathcal{A}| \log |\mathcal{A}|}.$$

The result in Proposition 9.5 follows directly from Proposition 9.4.

### 9.2.2 Stochastic multi-armed bandits

We now consider a different class of bandit algorithm, in which the cost function  $c_t$  is randomly sampled from some unknown underlying distribution. Equivalently, there is an action-dependent distribution  $P$  that governs the cost incurred by the agent whenever it executes an action. In other words, we can define a conditional probability distribution

$$P(c | a) \stackrel{\text{def}}{=} \mathbb{P}[c_t = c | a_t = a]$$

that governs the cost incurred by the agent as a function of the agent's action at each time step. This class of problems is known as *stochastic multi-armed bandit problem*, and it is significantly distinct from its adversarial counterpart.

For each action  $a \in \mathcal{A}$ , we define the expected cost associated with action  $a$  as the value

$$Q(a) = \mathbb{E}[c_t | a_t = a].$$

---

**Algorithm 9.5** The UCB algorithm (Auer, Cesa-Bianchi, and Fisher, 2002).

---

```

1:  $\hat{Q}(a) \leftarrow 0$  for all  $a \in \mathcal{A}$ ;
2:  $N(a) \leftarrow 0$  for all  $a \in \mathcal{A}$ ;
3: for  $t = 1, \dots, T$  do
4:   if  $t \leq |\mathcal{A}|$  then
5:     Select  $a_t = t$ 
6:   else
7:     Select

$$a_t = \operatorname{argmin}_{a \in \mathcal{A}} \left\{ \hat{Q}(a) - \sqrt{\frac{2 \log(t)}{N(a)}} \right\},$$

8:   end if
9:   Observe  $c_t(a_t)$ 
10:  Update

$$\hat{Q}(a) \leftarrow \hat{Q}(a) + \frac{\mathbb{I}[a_t] = a}{N(a) + 1} (c_t - \hat{Q}(a)).$$

11:   $N(a_t) \leftarrow N(a_t) + 1$ 
12: end for
```

---

In theory, we can experiment the different actions and use the corresponding costs to estimate  $Q(a)$  for all  $a \in \mathcal{A}$  (explore); once this process is complete, we can then use such estimates to guide the action selection (exploit). Once again, we must find a proper balance between the two.

Towards this goal, we adopt the principle of *optimism in the face of uncertainty*: besides the estimate of the average cost associated with each action, we also estimate a confidence interval for such costs. Then, at each time step  $t$ , we select the action with the smallest *lower confidence bound* (see Fig. 9.1). The approach thus obtained, known as UCB, is summarized in Algorithm 9.5, where we assume that the (finite) actions in  $\mathcal{A}$  are ordered as  $\mathcal{A} = \{1, 2, \dots, |\mathcal{A}|\}$ .

The next result provides a guarantee on the performance of UCB, providing a bound on the expected regret incurred by the algorithm when compared with that of the best action. Since now the costs are generated from the distribution  $P$ , the best action is well-defined, and we can rewrite the expected regret as

$$R_T = \sum_{t=1}^T \mathbb{E}_{\pi_t} [c_t] - TQ(a^*),$$

where

$$a^* \in \operatorname{argmin}_{a \in \mathcal{A}} Q(a)$$

and  $Q(a)$  is, as in Chapter 4,

$$Q(a) = \mathbb{E}_P [c].$$

**Theorem 9.6.** *After  $T$  time steps, the regret of Algorithm 9.5 verifies*

$$R_T \leq \sum_{a \neq a^*} \left[ \frac{8 \log(T)}{Q(a) - Q(a^*)} + \left(1 + \frac{\pi^2}{3}\right) (Q(a) - Q(a^*)) \right]$$

*Proof.* The proof rests on the fact that

$$R_T = \sum_{t=0}^{T-1} (\mathbb{E}_{\pi_t} [c_t] - Q(a^*)) = \sum_{a \in \mathcal{A}} (Q(a) - Q(a^*)) \mathbb{E}_{\pi_t} [N_T(a)].$$

By bounding the expected number of times that each (suboptimal) action is played, we can bound the regret incurred by the UCB1 algorithm. The complete proof is found in Section 9.3.  $\square$

### 9.3 Proofs

We provide the proofs of all results presented in the main text.

#### Proof of Proposition 9.2

The proof is essentially similar to that of Proposition 9.1. We write  $M_t$  to denote the expected number of “mistakes” incurred by the algorithm up to time step  $t$ , and let  $\varepsilon_t$  and  $\gamma_t$  denote the total weight of the “incorrect” and “correct” policies at time step  $t$ , respectively. Additionally, let

$$E_t = \frac{\varepsilon_t}{W_t}.$$

The value  $E_t$  corresponds to the probability of making a “mistake” at time step  $t$ . The expected total number of “mistakes” incurred by Algorithm 9.1 up to time step  $T$  is then given by

$$M_T = \sum_{t=0}^{T-1} E_t. \tag{9.12}$$

Now, at each time step  $t$ , after the weight update, we have that

$$W_{t+1} = (1 - \beta)\varepsilon_t + \gamma_t = W_t - \beta\varepsilon_t = W_t(1 - \beta E_t),$$

where the last equality follows from (9.12). Then,

$$W_T = W_0 \prod_{t=0}^{T-1} (1 - \beta E_t),$$

with  $W_0 = |\Pi|$ . On the other hand, let  $\pi^*$  denote the best policy in  $\Pi$  up to time step  $T$ , i.e.,

$$\pi^* = \operatorname{argmax}_{\pi \in \Pi} \{w_T(\pi)\},$$



and let  $m$  denote the number of mistakes incurred by  $\pi^*$  up to  $T$ . Then,

$$w_T(\pi^*) = (1 - \beta)^m \leq W_T \leq |\Pi| \prod_{t=0}^{T-1} (1 - \beta E_t).$$

Applying the logarithm to both sides, we have

$$m \log(1 - \beta) \leq \log |\Pi| + \sum_{t=0}^{T-1} \log(1 - \beta E_t).$$

Using (A.13) on the logarithms inside the summation yields

$$m \log(1 - \beta) \leq \log |\Pi| - \beta \sum_{t=0}^{T-1} E_t.$$

Using (9.12) and solving for  $M_T$ , we get

$$M_T \leq \frac{1}{\beta} \log |\Pi| - \frac{m}{\beta} \log(1 - \beta).$$

Finally, using (A.14),

$$M_T \leq \frac{1}{\beta} \log |\Pi| - m(1 + \beta).$$

The proof is complete by noting that  $R_T = M_T - m$ . □

### Proof of Proposition 9.3

We follow an argumentation similar to the previous proofs. For simplicity, we henceforth write  $c(\pi)$  to denote  $c_t(\pi(h_t))$ . We have that

$$w_t(\pi) = \exp \left\{ -\eta \sum_{t=0}^{T-1} c_t(\pi) \right\}.$$

Letting

$$W_t = \sum_{\pi \in \Pi} w_t(\pi),$$

we get

$$\begin{aligned} \log \left( \frac{W_T}{W_0} \right) &= \log \left( \sum_{\pi \in \Pi} e^{-\eta \sum_{t=0}^{T-1} c_t(\pi)} \right) - \log |\Pi| \\ &\geq \log \left( \max_{\pi \in \Pi} e^{-\eta \sum_{t=0}^{T-1} c_t(\pi)} \right) - \log |\Pi| \\ &= \max_{\pi \in \Pi} \log \left( e^{-\eta \sum_{t=0}^{T-1} c_t(\pi)} \right) - \log |\Pi| \\ &= -\eta \min_{\pi \in \Pi} \sum_{t=0}^{T-1} c_t(\pi) - \log |\Pi|. \end{aligned}$$

On the other hand, for any  $t = 0, \dots, T-1$ ,

$$\begin{aligned} \log \left( \frac{W_{t+1}}{W_t} \right) &= \log \left( \frac{\sum_{\pi \in \Pi} w_t(\pi) e^{-\eta c_{t+1}(\pi)}}{\sum_{\pi \in \Pi} w_t(\pi)} \right) \\ &= \log \mathbb{E}_{\pi \sim q_t} \left[ e^{-\eta c_{t+1}(\pi)} \right], \end{aligned}$$

where we denote by  $q_t$  the distribution over  $\Pi$  such that

$$q_t(\pi) = \frac{w_t(\pi)}{W_t}.$$

Using Hoeffding's lemma (A.39), we get

$$\log \left( \frac{W_{t+1}}{W_t} \right) \leq -\eta \mathbb{E}_{\pi \sim q_t} [c_{t+1}(\pi)] + \frac{\eta^2}{8}.$$

Adding for  $t = 0, \dots, T-1$ , yields

$$\sum_{t=0}^{T-1} \log \frac{W_{t+1}}{W_t} = \log \left( \frac{W_T}{W_0} \right) \leq -\eta \sum_{t=0}^{T-1} \mathbb{E}_{\pi \sim q_t} [c_t(\pi)] + \frac{\eta^2}{8} T.$$

Combining both inequalities for  $\log(W_T/W_0)$  we finally get

$$-\eta \min_{\pi \in \Pi} \sum_{t=0}^{T-1} c_t(\pi) - \log |\Pi| \leq -\eta \sum_{t=0}^{T-1} \mathbb{E}_{\pi \sim q_t} [c_t(\pi)] + \frac{\eta^2}{8} T$$

or, equivalently,

$$\sum_{t=0}^{T-1} \mathbb{E}_{\pi \sim q_t} [c_t(\pi)] - \min_{\pi \in \Pi} \sum_{t=0}^{T-1} c_t(\pi) \leq \frac{\log |\Pi|}{\eta} + \frac{\eta}{8} T.$$

The proof is complete.  $\square$

#### Proof of Proposition 9.4

The proof is similar to that of Proposition 9.3.

For simplicity of notation, let  $\pi(a)$  denote  $\pi(a \mid h_t)$  and define

$$W_t = \sum_{\pi \in \Pi} w_t(\pi), \quad q_t(\pi) = \frac{w_t(\pi)}{W_t}.$$

We use the following facts:

$$\sum_{\pi \in \Pi} q_t(\pi) \hat{c}_{\pi,t} = \sum_{\pi \in \Pi} q_t(\pi) \sum_{a \in \mathcal{A}} \pi(a) \hat{c}_t(a) = \sum_{a \in \mathcal{A}} p_t(a) \frac{c_t(a)}{p_t(a)} \mathbb{I}[a_t = a] = c_t(a_t); \quad (9.13)$$

$$\sum_{\pi \in \Pi} q_t(\pi) \hat{c}_{\pi,t}^2 = \sum_{\pi \in \Pi} q_t(\pi) \left( \sum_{a \in \mathcal{A}} \pi(a) \hat{c}_t(a) \right)^2 \leq \sum_{a \in \mathcal{A}} p_t(a) \frac{c_t^2(a)}{p_t^2(a)} \mathbb{I}[a_t = a] = \frac{c_t^2(a_t)}{p_t(a_t)}. \quad (9.14)$$

Following the same line of reasoning as in the proof of Proposition 9.3, we have

$$\begin{aligned} \log \left( \frac{W_T}{W_0} \right) &= \log \left( \sum_{\pi \in \Pi} e^{-\eta \sum_{t=0}^{T-1} \hat{c}_{\pi,t}} \right) - \log |\Pi| \\ &\geq \log \left( e^{-\eta \sum_{t=0}^{T-1} \hat{c}_{\pi,t}} \right) - \log |\Pi| \\ &= -\eta \sum_{t=0}^{T-1} \hat{c}_{\pi,t} - \log |\Pi|, \end{aligned}$$

for any  $\pi \in \Pi$ . On the other hand, for  $t = 0, \dots, T-1$ , we have

$$\frac{W_{t+1}}{W_t} = \sum_{\pi \in \Pi} \frac{w_t(\pi) e^{-\eta \hat{c}_{\pi,t}}}{\sum_{\pi \in \Pi} w_t(\pi)} = \sum_{\pi \in \Pi} q_t(\pi) e^{-\eta \hat{c}_{\pi,t}}.$$

From the Taylor series expansion,  $e^{-x} \leq 1 - x + \frac{1}{2}x^2$  for  $x \geq 0$ , yielding

$$\begin{aligned} \frac{W_{t+1}}{W_t} &\leq \sum_{\pi \in \Pi} q_t(\pi) \left( 1 - \eta \hat{c}_{\pi,t} + \frac{\eta^2}{2} \hat{c}_{\pi,t}^2 \right) \\ &\leq 1 + \sum_{\pi \in \Pi} q_t(\pi) \left( -\eta \hat{c}_{\pi,t} + \frac{\eta^2}{2} \hat{c}_{\pi,t}^2 \right). \end{aligned}$$

Computing the logarithm and using the fact that  $\log(x) \leq x - 1$ ,

$$\begin{aligned} \log \left( \frac{W_{t+1}}{W_t} \right) &\leq \sum_{\pi \in \Pi} q_t(\pi) \left( -\eta \hat{c}_{\pi,t} + \frac{\eta^2}{2} \hat{c}_{\pi,t}^2 \right) \\ &= -\eta c_t(a_t) - \frac{\eta^2}{2} \cdot \frac{c_t^2(a_t)}{p_t(a_t)}, \end{aligned}$$

where we used (9.13) and (9.14). Adding for all  $t = 0, \dots, T-1$ , yields

$$\log \left( \frac{W_T}{W_0} \right) \leq -\eta \sum_{t=0}^{T-1} c_t(a_t) - \frac{\eta^2}{2} \sum_{t=0}^{T-1} \frac{c_t^2(a_t)}{p_t(a_t)}.$$

We can now combine both inequalities for  $\log(W_T/W_0)$  to get

$$-\eta \sum_{t=0}^{T-1} \hat{c}_{\pi,t} - \log |\Pi| \leq -\eta \sum_{t=0}^{T-1} c_t(a_t) - \frac{\eta^2}{2} \sum_{t=0}^{T-1} \frac{c_t^2(a_t)}{p_t(a_t)}.$$

or, equivalently,

$$\sum_{t=0}^{T-1} c_t(a_t) - \sum_{t=0}^{T-1} \hat{c}_{\pi,t} \leq \frac{\log |\Pi|}{\eta} - \frac{\eta}{2} \sum_{t=0}^{T-1} \frac{c_t^2(a_t)}{p_t(a_t)}.$$

We now note that

$$\begin{aligned}
\mathbb{E} \left[ \sum_{t=0}^{T-1} \hat{c}_{\pi,t} \right] &= \mathbb{E} \left[ \sum_{t=0}^{T-1} \mathbb{E} [\hat{c}_{\pi,t} \mid \mathbf{a}_0, \dots, \mathbf{a}_{t-1}] \right] \\
&= \mathbb{E} \left[ \sum_{t=0}^{T-1} \sum_{a \in \mathcal{A}} \pi(a) \mathbb{E} [\hat{c}_t(a) \mid \mathbf{a}_0, \dots, \mathbf{a}_{t-1}] \right] \\
&= \mathbb{E} \left[ \sum_{t=0}^{T-1} \sum_{a \in \mathcal{A}} \pi(a) \frac{c_t(a)}{p_t(a)} \mathbb{E} [\mathbb{I}[\mathbf{a}_t = a] \mid \mathbf{a}_0, \dots, \mathbf{a}_{t-1}] \right] \\
&= \mathbb{E} \left[ \sum_{t=0}^{T-1} \sum_{a \in \mathcal{A}} \pi(a) \frac{c_t(a)}{p_t(a)} p_t(a) \right] \\
&= \mathbb{E} \left[ \sum_{t=0}^{T-1} c_{\pi,t} \right]
\end{aligned}$$

and that  $\mathbb{E} \left[ \frac{c_t^2(\mathbf{a}_t)}{p_t(\mathbf{a}_t)} \right] \leq |\mathcal{A}|$ , finally yielding

$$\max \sum_{t=0}^{T-1} \mathbb{E} [c_t(\mathbf{a}_t) - c_{\pi,t}] \stackrel{\text{def}}{=} R_T \leq \frac{\log |\Pi|}{\eta} - \frac{\eta}{2} |\mathcal{A}| T.$$

The proof is complete.  $\square$

### Proof of Theorem 9.6

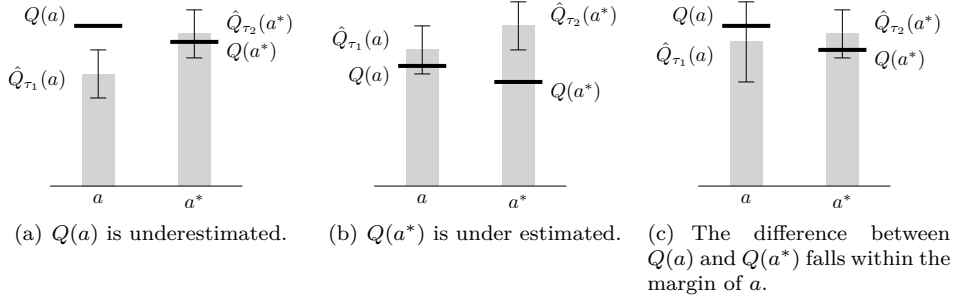
We bound the term  $\mathbb{E}_{\pi_t} [N_T(a)]$ . Let  $C_{N,t}$  denote the confidence bound used in Algorithm 9.5, i.e.,

$$C_{N,t} = \sqrt{\frac{2 \log(t)}{N}},$$

and let  $\ell$  be any positive integer. After the initial rounds, where each action is played once, UCB selects the action with the lowest confidence bound. Then, for any  $a \neq a^*$ ,

$$\begin{aligned}
N_T(a) &= 1 + \sum_{t=|\mathcal{A}|+1}^{T-1} \mathbb{I}[\mathbf{a}_t = a] \\
&\leq \ell + \sum_{t=|\mathcal{A}|+1}^{T-1} \mathbb{I}[\mathbf{a}_t = a, N_t(a) \geq \ell] \\
&\leq \ell + \sum_{t=|\mathcal{A}|+1}^{T-1} \mathbb{I} \left[ \min_{\ell < \tau_1 < t} \hat{Q}_{\tau_1}(a) - C_{\tau_1,t}(a) \leq \max_{0 < \tau_2 < t} \hat{Q}_{\tau_2}(a^*) - C_{\tau_2,t}(a^*) \right] \\
&\leq \ell + \sum_{t=|\mathcal{A}|+1}^{T-1} \sum_{\tau_1=\ell}^t \sum_{\tau_2=0}^t \mathbb{I} \left[ \hat{Q}_{\tau_1}(a) - C_{\tau_1,t}(a) \leq \hat{Q}_{\tau_2}(a^*) - C_{\tau_2,t}(a^*) \right].
\end{aligned}$$

If  $\hat{Q}_{\tau_1}(a) - C_{\tau_1,t}(a) \leq \hat{Q}_{\tau_2}(a^*) - C_{\tau_2,t}(a^*)$ , then at least one of three situations must necessarily take place (see Fig. 9.2):



**Figure 9.2** Illustration of the three possible situations that may occur when  $\hat{Q}_{\tau_1}(a) - C_{\tau_1,t}(a) \leq \hat{Q}_{\tau_2}(a^*) - C_{\tau_2,t}(a^*)$ .

- The agent underestimates the cost of action  $a$ , i.e.,

$$\hat{Q}_{\tau_1}(a) + C_{\tau_1,t} \leq Q(a) \quad (9.15)$$

- The agent overestimates the cost of action  $a^*$ , i.e.,

$$\hat{Q}_{\tau_2}(a^*) - C_{\tau_2,t} \geq Q(a^*). \quad (9.16)$$

- The difference between the true values of  $a$  and  $a^*$  falls within the confidence margin, i.e.,

$$Q(a^*) - Q(a) \leq 2C_{\tau_1,t}. \quad (9.17)$$

Let us consider each situation in particular. For (9.15), using Hoeffding's inequality (equation A.40 in page 416) we get

$$\mathbb{P} \left[ Q(a) - \hat{Q}_{\tau_1}(a) \geq C_{\tau_1,t} \right] \leq e^{-2\tau_1 C_{\tau_1,t}} = t^{-4}.$$

Similarly, for (9.16), we get

$$\mathbb{P} \left[ \hat{Q}_{\tau_2}(a^*) - Q(a^*) \geq C_{\tau_2,t} \right] \leq e^{-2\tau_2 C_{\tau_2,t}} = t^{-4}.$$

Finally, in (9.17), we get that

$$\sqrt{\frac{2 \log(t)}{\tau_1}} \geq \frac{1}{2}(Q(a^*) - Q(a)).$$

or, equivalently,

$$\tau_1 \leq \frac{8 \log(t)}{(Q(a^*) - Q(a))^2}.$$

Setting

$$\ell = \left\lceil \frac{8 \log(t)}{(Q(a^*) - Q(a))^2} \right\rceil,$$

we can observe that (9.17) is not possible. Computing the expectation of  $N_T(a)$ , we thus get

$$\begin{aligned}
& \mathbb{E}_{\pi_t} [N_T(a)] \\
& \leq \ell + \sum_{t=|\mathcal{A}|+1}^{T-1} \sum_{\tau_1=\ell}^t \sum_{\tau_2=0}^t \mathbb{P}_{\pi_t} \left[ \hat{Q}_{\tau_1}(a) - C_{\tau_1,t}(a) \leq \hat{Q}_{\tau_2}(a^*) - C_{\tau_2,t}(a^*) \right] \\
& \leq \ell + \sum_{t=|\mathcal{A}|+1}^{T-1} \sum_{\tau_1=\ell}^t \sum_{\tau_2=0}^t \left( \mathbb{P} \left[ Q(a) - \hat{Q}_{\tau_1}(a) \geq C_{\tau_1,t} \right] + \mathbb{P} \left[ \hat{Q}_{\tau_2}(a^*) - Q(a^*) \geq C_{\tau_2,t} \right] \right) \\
& \leq \ell + \sum_{t=|\mathcal{A}|+1}^{T-1} \sum_{\tau_1=\ell}^t \sum_{\tau_2=0}^t 2t^{-4} \\
& \leq \ell + \sum_{t=|\mathcal{A}|+1}^{T-1} 2t^{-2}.
\end{aligned}$$

The summation can be bounded above by  $\pi^2/3$ ,<sup>3</sup> yielding

$$\mathbb{E}_{\pi_t} [N_T(a)] \leq \ell + \frac{\pi^2}{3} = \frac{8 \log(t)}{(Q(a^*) - Q(a))^2} + 1 + \frac{\pi^2}{3}.$$

The conclusion follows.  $\square$

## 9.4 Exercises

### Exercise 9.1.

Using the weighted majority algorithm with  $\beta = \frac{1}{2}$ , compute the weights associated with the five predictors in Example 9.1 after day 5, knowing that the weather was always sunny except on Day 3.

### Exercise 9.2.

Compute the actual number of mistakes incurred by the weighted majority in the conditions of Question 9.1.

### Exercise 9.3.

Work out the details of the proof of Proposition 9.5.

---

<sup>3</sup>Euler showed, in 1741, that  $\sum_{t=0}^{\infty} t^{-2} = \frac{\pi}{6}$ , solving what was known as the *Basel problem*.