6 The Explore-Then-Commit Algorithm

The first bandit algorithm of the book is called **explore-then-commit** (ETC), which explores by playing each arm a fixed number of times and then exploits by committing to the arm that appeared best during exploration.

For this chapter, as well as Chapters 7 to 9, we assume that all bandit instances are in $\mathcal{E}^k_{\text{SG}}(1)$, which means the reward distribution for all arms is 1-subgaussian.

The focus on subgaussian distributions is mainly for simplicity. Many of the techniques in the chapters that follow can be applied to other stochastic bandits such as those listed in Table 4.1. The key difference is that new concentration analysis is required that exploits the different assumptions. The Bernoulli case is covered in Chapter 10, where other situations are discussed along with references to the literature. Notice that the subgaussian assumption restricts the subgaussian constant to $\sigma = 1$, which saves us from endlessly writing σ. All results hold for other subgaussian constants by scaling the rewards (see Lemma 5.4). Two points are obscured by this simplification:

(a) All the algorithms that follow rely on the knowledge of σ.

(b) It may happen that P_i is subgaussian for all arms, but with a different subgaussian constant for each arm. Algorithms are easily adapted to this situation if the subgaussian constants are known, as you will investigate in Exercise 7.2. The situation is more complicated when the subgaussian constant is unknown (Exercise 7.7).

6.1 Algorithm and Regret Analysis

ETC is characterised by the number of times it explores each arm, denoted by a natural number m. Because there are k actions, the algorithm will explore for mk rounds before choosing a single action for the remaining rounds. Let $\hat{\mu}_i(t)$ be the

average reward received from arm i after round t, which is written formally as

$$\hat{\mu}_i(t) = \frac{1}{T_i(t)} \sum_{s=1}^{t} \mathbb{I}\{A_s = i\} X_s \,,$$

where $T_i(t) = \sum_{s=1}^{t} \mathbb{I}\{A_s = i\}$ is the number of times action i has been played after round t. The ETC policy is given in Algorithm 1 below.

1: **Input** m.
2: In round t choose action

$$A_t = \begin{cases} (t \bmod k) + 1 \,, & \text{if } t \leq mk \,; \\ \operatorname{argmax}_i \hat{\mu}_i(mk) \,, & t > mk \,. \end{cases}$$

(ties in the argmax are broken arbitrarily)

Algorithm 1: Explore-then-commit.

Recall that μ_i is the mean reward when playing action i and $\Delta_i = \mu^* - \mu_i$ is suboptimality gap between the mean of action i and the optimal action.

THEOREM 6.1. *When ETC is interacting with any* 1*-subgaussian bandit and* $1 \leq m \leq n/k$*,*

$$R_n \leq m \sum_{i=1}^{k} \Delta_i + (n - mk) \sum_{i=1}^{k} \Delta_i \exp\left(-\frac{m\Delta_i^2}{4}\right) .$$

Proof Assume without loss of generality that the first arm is optimal, which means that $\mu_1 = \mu^* = \max_i \mu_i$. By the decomposition given in Lemma 4.5, the regret can be written as

$$R_n = \sum_{i=1}^{k} \Delta_i \mathbb{E}\left[T_i(n)\right] . \tag{6.1}$$

In the first mk rounds, the policy is deterministic, choosing each action exactly m times. Subsequently it chooses a single action maximising the average reward during exploration. Thus,

$$\begin{aligned} \mathbb{E}\left[T_i(n)\right] &= m + (n - mk)\mathbb{P}\left(A_{mk+1} = i\right) \\ &\leq m + (n - mk)\mathbb{P}\left(\hat{\mu}_i(mk) \geq \max_{j \neq i} \hat{\mu}_j(mk)\right) . \end{aligned} \tag{6.2}$$

The probability on the right-hand side is bounded by

$$\begin{aligned} \mathbb{P}\left(\hat{\mu}_i(mk) \geq \max_{j \neq i} \hat{\mu}_j(mk)\right) &\leq \mathbb{P}\left(\hat{\mu}_i(mk) \geq \hat{\mu}_1(mk)\right) \\ &= \mathbb{P}\left(\hat{\mu}_i(mk) - \mu_i - (\hat{\mu}_1(mk) - \mu_1) \geq \Delta_i\right) . \end{aligned}$$

The next step is to check that $\hat{\mu}_i(mk) - \mu_i - (\hat{\mu}_1(mk) - \mu_1)$ is $\sqrt{2/m}$-subgaussian,

which by the properties of subgaussian random variables follows from the definitions of $(\hat{\mu}_j)_j$ and the algorithm. Hence by Corollary 5.5,

$$\mathbb{P}\left(\hat{\mu}_i(mk) - \mu_i - \hat{\mu}_1(mk) + \mu_1 \geq \Delta_i\right) \leq \exp\left(-\frac{m\Delta_i^2}{4}\right). \tag{6.3}$$

Substituting Eq. (6.3) into Eq. (6.2) and the regret decomposition (Eq. (6.1)) gives the result. □

The bound in Theorem 6.1 illustrates the trade-off between exploration and exploitation. If m is large, then the policy explores for too long, and the first term will be large. On the other hand, if m is too small, then the probability that the algorithm commits to the wrong arm will grow, and the second term becomes large. The question is how to choose m. Assume that $k = 2$ and that the first arm is optimal so that $\Delta_1 = 0$, and abbreviate $\Delta = \Delta_2$. Then the bound in Theorem 6.1 simplifies to

$$R_n \leq m\Delta + (n - 2m)\Delta \exp\left(-\frac{m\Delta^2}{4}\right) \leq m\Delta + n\Delta \exp\left(-\frac{m\Delta^2}{4}\right). \tag{6.4}$$

For large n the quantity on the right-hand side of Eq. (6.4) is minimised up to a possible rounding error by

$$m = \max\left\{1, \left\lceil \frac{4}{\Delta^2} \log\left(\frac{n\Delta^2}{4}\right)\right\rceil\right\}, \tag{6.5}$$

and for this choice and any n, the regret is bounded by

$$R_n \leq \min\left\{n\Delta, \Delta + \frac{4}{\Delta}\left(1 + \max\left\{0, \log\left(\frac{n\Delta^2}{4}\right)\right\}\right)\right\}. \tag{6.6}$$

In Exercise 6.2 you will show that Eq. (6.6) implies that

$$R_n \leq \Delta + C\sqrt{n}\,, \tag{6.7}$$

where $C > 0$ is a universal constant. In particular, when $\Delta \leq 1$ as is often assumed, we get

$$R_n \leq 1 + C\sqrt{n}\,,$$

Bounds of this type are called **worst-case**, **problem free** or **problem independent** (see Eq. (4.2) or Eq. (4.3)). The reason is that the bound only depends on the horizon and class of bandits for which the algorithm is designed, and not the specific instance within that class. Because the suboptimality gap does not appear, bounds like this are sometimes called **gap-free**. In contrast, bounds like the one in Eq. (6.6) are called **gap/problem/distribution/instance dependent**.

Note that without the condition $\Delta \leq 1$, the worst-case bound for ETC is infinite. In fact, without a bound on the reward range, the worst-case bound of all reasonable algorithms (that try each action at least once) will also be infinite. With the understanding that Eq. (6.7) gives rise to a meaningful worst-case

bound for bandits with bounded reward range, we take the liberty and will also call bounds like that in Eq. (6.7) a worst-case bound.

The bound in (6.6) is close to optimal (see Part IV), but there is a caveat. The choice of m that defines the policy and leads to this bound depends on both the suboptimality gap and the horizon. While the horizon is sometimes known in advance, it is seldom reasonable to assume knowledge of the suboptimality gap. You will show in Exercise 6.5 that there is a choice of m depending only on n, for which $R_n = O(n^{2/3})$ regardless of the value of Δ. Alternatively, the number of plays before commitment can be made data dependent, which means the learner plays arms alternately until it decides based on its observations to commit to a single arm for the remainder (Exercise 6.5). ETC also has the property that its immediate expected regret per time step is monotonically decreasing as time goes by, though not in a nice smooth fashion. This monotone decreasing property is a highly desirable property. In later chapters we will see policies where the decrease is smoother.

EXPERIMENT 6.1 Fig. 6.1 shows the expected regret of ETC when playing a Gaussian bandit with $k = 2$ and means $\mu_1 = 0$ and $\mu_2 = -\Delta$. The horizon is set to $n = 1000$, and the suboptimality gap Δ is varied between 0 and 1. Each data point is the average of 10^5 simulations, which makes the error bars invisible. The results show that the theoretical upper bound provided by Theorem 6.1 is quite close to the actual performance.

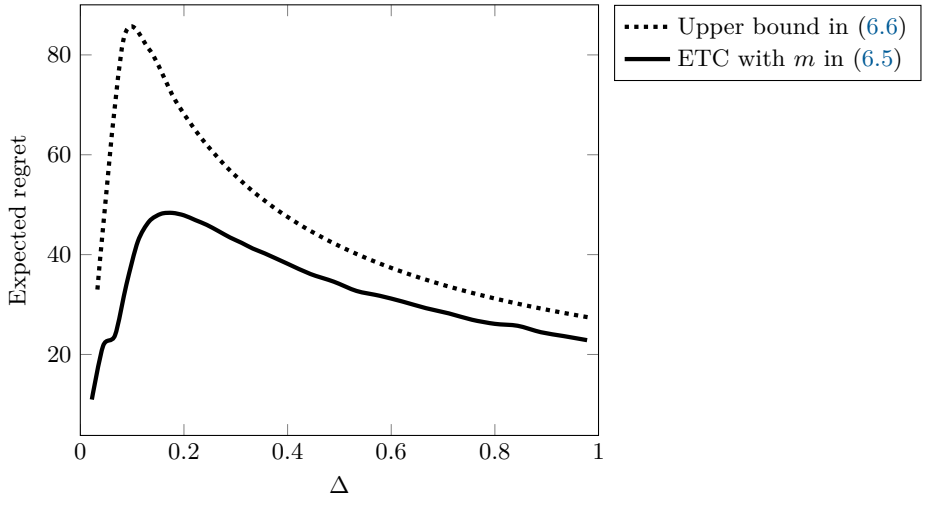

Figure 6.1 The expected regret of ETC and the upper bound in Eq. (6.6).