Reinforcement Learning

Mohammed Luqmaan Akhtar

February 2026

Exploration–Exploitation and Action-Value Methods in Multi-Armed Bandits

A k-armed bandit problem models a scenario in which an agent repeatedly selects one of k possible actions and receives a reward drawn from an unknown distribution associated with that action. At each timestep t, the agent selects an action $A_t \in \{1, \ldots, k\}$ and observes reward R_t.

The objective is to maximize the expected cumulative reward:

$$\mathbb{E}\left[\sum_{t=1}^{T} R_t\right]$$

Exploration–Exploitation Dilemma

A central challenge in bandit problems is the exploration–exploitation tradeoff. **Exploration** refers to selecting actions to gain information about their reward distributions, while **exploitation** refers to selecting the action currently believed to yield the highest reward. The dilemma is that the agent has to choose one. If he chooses to explore he gains the opportunity to get a larger reward long term but at the same time loses short term. If he chooses to exploit then he loses the opportunity to make a gain long term. This dilemma is what we need to ”solve”.

Upper Confidence Bound (UCB) Algorithm

The Upper Confidence Bound (UCB) algorithm addresses the exploration–exploitation dilemma using the principle of optimism in the face of uncertainty. Actions with high uncertainty are assigned an exploration bonus, encouraging their selection.

At timestep t, UCB selects the action:

$$A_t = \arg\max_a \left[Q_t(a) + c\sqrt{\frac{\ln t}{N_t(a)}}\right]$$

where:

- $Q_t(a)$ is the estimated value of action a
- $N_t(a)$ is the number of times action a has been selected before time t
- $c > 0$ controls exploration strength

The first term promotes exploitation of high-value actions, while the second term promotes exploration of less-sampled actions. As $N_t(a)$ increases, the uncertainty term shrinks, gradually favoring exploitation.

Optimistic Initialization Strategy

Optimistic initialization is a simple method for encouraging exploration in greedy action selection. Instead of initializing action-value estimates to zero, they are initialized to an artificially high value:

$$Q_1(a) = Q_{\text{init}} \quad \forall a$$

where Q_{init} exceeds plausible reward values.

Because all actions initially appear highly rewarding, the agent is forced to try each action. As rewards are observed, estimates decrease toward true values, naturally shifting behavior from exploration to exploitation without requiring randomness.

Incremental Update Methods for Action-Value Estimates

The true value of an action is defined as the expected reward:

$$q(a) = \mathbb{E}[R_t \mid A_t = a]$$

A natural estimator is the sample average of observed rewards:

$$Q_n(a) = \frac{1}{n} \sum_{i=1}^{n} R_i$$

To avoid storing all past rewards, this estimate can be updated incrementally. The incremental mean update is:

$$Q_{n+1}(a) = Q_n(a) + \frac{1}{n} \left(R_n - Q_n(a) \right)$$

This form reveals that the estimate moves toward the new reward by a fraction $1/n$ of the prediction error.

In non-stationary environments where reward distributions change over time, a constant step-size update is preferred:

$$Q_{n+1}(a) = Q_n(a) + \alpha \left(R_n - Q_n(a) \right)$$

where $0 < \alpha \leq 1$ is a fixed learning rate. This produces an exponential recency-weighted average, allowing the estimate to track changing action values.