Module 6: Lesson 1

Reinforcement Learning: Multi-Armed Bandits

Outline

- Reinforcement Learning: Using experience to learn
- Introduction to Multi-Armed Bandits

Reinforcement Learning: Using experience to learn

In this module, we deal with one of the essential ingredients of RL algorithms: the lack of knowledge or full modelization of the transitions between states.

In the previous module, Dynamic Programming assumed perfect knowledge about how the environment transitions from one state to another. In many cases, it is unfeasible or impractical to obtain precise estimates of the transitions.

The most straightforward application of learning from unknown environments is Multi-Armed Bandits. In Module 7, we will combine all our accumulated knowledge to develop fully-fledged RL algorithms and implementations.

Multi-Armed Bandits: The value of actions

Consider the following learning situation. You repeatedly face a choice among k different options or actions.

The objective is to maximize the expected total reward over some time period where you select some actions.

- An analogy is that of an investor choosing between stocks: Each action is the selection of a stock, and each reward is the return from the investment.
- Through experience, the investor may be able to learn which stocks tend to generate higher returns and select them appropriately.

Each of the k actions has an expected reward given that that action is selected; this is the *value* of that action.

At a time step t, the value of an arbitrary action a, denoted $Q^*(a)$, is the expected reward r_t given that a is selected:

$$Q^*(a) = E\{r_t | a_t = a\} \tag{1}$$

We never know the action values with certainty, although we may have more or less precise estimates.

Multi-Armed Bandits: Choosing actions

We denote the estimated value of action a at time step t as $Q_t(a)$:

$$Q_t(a) = \frac{\sum_{s=1}^{t-1} r_t \mathbf{1}(a_s = a)}{\sum_{s=1}^{t-1} \mathbf{1}(a_s = a)} \tag{2}$$

That is, $Q_t(a)$ represents the average reward obtained when a was chosen.

The simplest action selection rule is to select one of the actions with the highest estimated value: a *greedy* policy.

$$A_t = \arg\max_a Q_t(a) \tag{3}$$

Greedy action selection always exploits current knowledge to maximize immediate reward but reduces the extent of exploration to learn about the rewards of other alternatives.

Near-greedy action selection: ε-greedy methods. These behave greedily most of the time, but with some probability ε, they select randomly from among all the actions with equal probability.

Multi-Armed Bandits: Updating the value of actions

Given the estimated reward $Q_t(a)$ and an observed reward after choosing a, r_t, the new average of all rewards can be computed as:

$$Q_t(a) = Q_{t-1}(a) + \frac{1}{N_t(a)}[r_t - Q_{t-1}(a)] \tag{4}$$

We often encounter reinforcement learning problems that are nonstationary: the underlying reward process changes over time. In such cases, it makes sense to give more weight to recent rewards than to distant rewards:

$$Q_t(a) = Q_{t-1}(a) + \alpha[r_t - Q_{t-1}(a)] \tag{5}$$

where $\alpha \in (0, 1)$ is a constant step-size parameter.

Summary of Lesson 1

In Lesson 1, we have looked at:

- How Reinforcement Learning obtains updates of optimal policies by experience
- How to select actions and update their values in Multi-Armed Bandit problems

⇒ **References for this Lesson:**

Sutton, Richard S., Barto, Andrew G. *Reinforcement Learning: An Introduction*. MIT Press, 2018. (see Chapter 2)

TO DO NEXT: Now, please go to the associated Jupyter Notebook for this lesson to get further insights on the use of RL to devise optimal actions from experience without knowledge of the setup.

In the next lesson, we will cover a practice example of a k-Bandit Problem in a stationary setup.