

# Chapter 2: Multi-armed Bandits - Highlights

Jacob Taylor Cassidy

November 26, 2022

- The most important feature distinguishing reinforcement learning from other types of learning is that it uses training information that *evaluates* the actions taken rather than *instructs* by giving correct actions.
- Purely evaluative feedback indicates how good the action taken was, but not whether it was the best or the worst action possible.
- one that does not involve learning to act in more than one situation. This *nonassociative* setting
- *associative*, that is, when actions are taken in more than one situation.

## 1 A k-armed Bandit Problem

- You are faced repeatedly with a choice among  $k$  different options, or actions. After each choice you receive a numerical reward chosen from a stationary probability distribution that depends on the action you selected. Your objective is to maximize the expected total reward over some time period, for example, over 1000 action selections, or *time steps*.
- This is the original form of the *k-armed bandit problem*, so named by analogy to a slot machine, or “one-armed bandit,” except that it has  $k$  levers instead of one.
- Through repeated action selections you are to maximize your winnings by concentrating your actions on the best levers.
- Today the term “bandit problem” is sometimes used for a generalization of the problem described above, but in this book we use it to refer just to this simple case.
- In our  $k$ -armed bandit problem, each of the  $k$  actions has an expected or mean reward given that action is selected; let us call this the *value* of that action. We denote the action selected on time step  $t$  as  $A_t$ , and the corresponding reward as  $R_t$ . The value then of an arbitrary action  $a$ , denoted  $q_*(a)$ , is the expected reward given that  $a$  is selected:

$$q_*(a) \doteq \mathbb{E}[R_t | A_t = a]$$

- We assume that you do not know the action values with certainty, although you may have estimates. We denote the estimated value of action  $a$  at time step  $t$  as  $Q_t(a)$ . We would like  $Q_t(a)$  to be close to  $q_*(a)$ .
- If you maintain estimates of the action values, then at any time step there is at least one action whose estimated value is greatest. We call these the *greedy* actions. When you select one of these actions, we say that you are *exploiting* your current knowledge of the values of the actions. If instead you select one of the nongreedy actions, then we say you are *exploring*, because this enables you to improve your estimate of the nongreedy action's value.
- Reward is lower in the short run, during exploration, but higher in the long run because after you have discovered the better actions, you can exploit them many times.

## 2 Action-value Methods

- methods for estimating the values of actions and for using the estimates to make action selection decisions, which we collectively call *action-value methods*.

•

$$Q_t(a) \doteq \frac{\text{sum of rewards when } a \text{ taken prior to } t}{\text{number of times } a \text{ taken prior to } t} = \frac{\sum_{i=1}^{t-1} R_i \cdot \mathbb{1}_{A_i=a}}{\sum_{i=1}^{t-1} \mathbb{1}_{A_i=a}} \quad (2.1)$$

where  $\mathbb{1}_{\text{predicate}}$  denotes the random variable that is 1 if *predicate* is true and 0 if it is not. If the denominator is zero, then we instead define  $Q_t(a)$  as some default value, such as 0. As the denominator goes to infinity, by the law of large numbers,  $Q_t(a)$  converges to  $q_*(a)$ . We call this the *sample-average* method for estimating action values because each estimate is an average of the sample of relevant rewards.

- The simplest action selection rule is to select one of the actions with the highest estimated value, that is, one of the greedy actions as defined in the previous section.
- We write this greedy action selection method as

$$A_t \doteq \underset{a}{\operatorname{argmax}} Q_t(a)$$

- A simple alternative is to behave greedily most of the time, but every once in a while, say with small probability  $\epsilon$ , instead select randomly from among all the actions with equal probability, independently of the action-value estimates. We call methods using this near-greedy action selection rule  $\epsilon$ -greedy methods.

- An advantage of these methods is that, in the limit as the number of steps increases, every action will be sampled an infinite number of times, thus ensuring that all the  $Q_t(a)$  converge to  $q_*(a)$ . This of course implies that the probability of selecting the optimal action converges to greater than  $1 - \epsilon$ , that is, to near certainty.

### 3 The 10-armed Testbed

- .

### 4 Incremental Implementation

- .

### 5 Tracking a Nonstationary Problem

- .

### 6 Optimistic Initial Values

- .

### 7 Upper-Confidence-Bound Action Selection

- .

### 8 Gradient Bandit Algorithms

- .

### 9 Associative Search (Contextual Bandits)

- .

### 10 Summary

- .