

Chapter 2 - Multi-armed Bandits Answers

Stéphane Liem NGUYEN

August 3, 2021

Exercises with (*corrected*) were corrected based on the [Errata](#).
These are my own answers and mistakes or errors are possible.

Exercise 2.1 (p. 28) In ϵ -greedy action selection, for the case of two actions and $\epsilon = 0.5$, what is the probability that the greedy action is selected?

Recall that at each time step, ϵ -greedy action selection methods select with probability ϵ a random action uniformly out of k actions (if there's k actions) and with probability $1 - \epsilon$ one of the greedy actions with ties broken arbitrarily (for instance randomly).

From the recall, we can say that the probability of the greedy action to be selected is $1 - \epsilon = 0.5$.

Exercise 2.2: Bandit example (p. 30) Consider a k -armed bandit problem with $k = 4$ actions, denoted 1, 2, 3, and 4. Consider applying to this problem a bandit algorithm using ϵ -greedy action selection, sample-average action-value estimates, and initial estimates of $Q_1(a) = 0$, for all a . Suppose the initial sequence of actions and rewards is $A_1 = 1, R_1 = -1, A_2 = 2, R_2 = 1, A_3 = 2, R_3 = -2, A_4 = 2, R_4 = 2, A_5 = 3, R_5 = 0$. On some of these time steps the ϵ case may have occurred, causing an action to be selected at random. On which time steps did this definitely occur? On which time steps could this possibly have occurred?

We already recalled what was the ϵ -greedy action selection methods. We can recall quickly that sample-average action-value estimates compute estimates $Q_t(a)$ based on an empirical mean of the rewards received when the action a is taken. The empirical means can be computed incrementally as explained in the book.

To know in what time step did the ϵ case definitely occur, we use the information that the decision-maker only picks a non-greedy action (not actions with highest current estimates) when exploring.

For this reason, we also compute estimates after each action-selection:

1. Take action 1, receive reward -1 and update estimates with $Q_2(1) = -1$ and keep $Q_1(a) = 0, \forall a \neq 1$. For the next action selection, the action 1 cannot be selected from greedy selection because its estimate is lower than the rest. The action 1 can only be picked from exploration for the next action selection.

2. Take action 2, receive reward 1 and update estimates with $Q_2(2) = 1$ and keep $Q_2(1) = -1$, $Q_1(a) = 0$, $\forall a \notin \{1, 2\}$. For the next time step, if we pick another action than 2 then it would be due to exploration.
3. Take action 2, receive reward -2 and update estimates with $Q_3(2) = \frac{1+(-2)}{2} = -0.5$ and keep $Q_2(1) = -1$, $Q_1(a) = 0$, $\forall a \notin \{1, 2\}$. For the next action selection, the action 2 can only be picked from exploration because its estimate is lower than the estimates of actions 3 and 4.
4. Take action 2, receive reward 2 and update estimates with $Q_4(2) = \frac{1+(-2)+2}{3} = \frac{1}{3}$ and keep $Q_2(1) = -1$, $Q_1(a) = 0$, $\forall a \notin \{1, 2\}$. For the next time step, if we pick another action than 2 then it would be due to exploration.
5. Take action 3, receive reward 0 and update estimates with $Q_2(3) = 0$ and keep $Q_4(2) = \frac{1}{3}$, $Q_2(1) = -1$, $Q_1(a) = 0$, $\forall a \notin \{1, 2, 3\}$.

The ϵ case could have occurred at all time steps but it definitely occurred for time step 4 and time step 5.

Exercise 2.3 (p. 30) In the comparison shown in Figure 2.2, which method will perform best in the long run in terms of cumulative reward and probability of selecting the best action? How much better will it be? Express your answer quantitatively.

Exercise 2.4 (p. 33) If the step-size parameters, α_n , are not constant, then the estimate Q_n is a weighted average of previously received rewards with a weighting different from that given by (2.6). What is the weighting on each prior reward for the general case, analogous to (2.6), in terms of the sequence of step-size parameters?

Exercise 2.5 (programming) (p. 33) Design and conduct an experiment to demonstrate the difficulties that sample-average methods have for nonstationary problems. Use a modified version of the 10-armed testbed in which all the $q_*(a)$ start out equal and then take independent random walks (say by adding a normally distributed increment with mean 0 and standard deviation 0.01 to all the $q_*(a)$ on each step). Prepare plots like Figure 2.2 for an action-value method using sample averages, incrementally computed, and another action-value method using a constant step-size parameter, $\alpha = 0.1$. Use $\epsilon = 0.1$ and longer runs, say of 10,000 steps.

Exercise 2.6: Mysterious Spikes (p. 35) The results shown in Figure 2.3 should be quite reliable because they are averages over 2000 individual, randomly chosen 10-armed bandit tasks. Why, then, are there oscillations and spikes in the early part of the curve for the optimistic method? In other words, what might make this method perform particularly better or worse, on average, on particular early steps?

Exercise 2.7: Unbiased Constant-Step-Size Trick (corrected) (p. 35)

In most of this chapter we have used sample averages to estimate action values because sample averages do not produce the initial bias that constant step sizes do (see the analysis leading to (2.6)). However, sample averages are not a completely satisfactory solution because they may perform poorly on nonstationary problems. Is it possible to avoid the bias of constant step sizes while retaining their advantages on nonstationary problems? One way is to use a step size of

$$\beta_n \doteq \alpha / \bar{o}_n \quad (1)$$

to process the n th reward for a particular action, where $\alpha > 0$ is a conventional constant step size, and \bar{o}_n is a trace of one that starts at 0:

$$\bar{o}_n \doteq \bar{o}_{n-1} + \alpha \cdot (1 - \bar{o}_{n-1}), \quad \text{for } n > 0, \quad \text{with } \bar{o}_0 \doteq 0 \quad (2)$$

Carry out an analysis like that in (2.6) to show that Q_n is an exponential recency-weighted average *without initial bias*.

Exercise 2.8: UCB Spikes (p. 36) In Figure 2.4 the UCB algorithm shows a distinct spike in performance on the 11th step. Why is this? Note that for your answer to be fully satisfactory it must explain both why the reward increases on the 11th step and why it decreases on the subsequent steps. Hint: If $c = 1$, then the spike is less prominent.

Exercise 2.9 (p. 37) Show that in the case of two actions, the soft-max distribution is the same as that given by the logistic, or sigmoid, function often used in statistics and artificial neural networks.

Exercise 2.10 (p. 41) Suppose you face a 2-armed bandit task whose true action values change randomly from time step to time step. Specifically, suppose that, for any time step, the true values of actions 1 and 2 are respectively 10 and 20 with probability 0.5 (case A), and 90 and 80 with probability 0.5 (case B). If you are not able to tell which case you face at any step, what is the best expected reward you can achieve and how should you behave to achieve it? Now suppose that on each step you are told whether you are facing case A or case B (although you still don't know the true action values). This is an associative search task. What is the best expected reward you can achieve in this task, and how should you behave to achieve it?

Exercise 2.11 (programming) (p. 44) Make a figure analogous to Figure 2.6 (parameter study) for the nonstationary case outlined in **Exercise 2.5**. Include the constant-step-size ϵ -greedy algorithm with $\alpha = 0.1$. Use runs of 200,000 steps and, as a performance measure for each algorithm and parameter setting, use the average reward over the last 100,000 steps.