Solutions for Chapter 2 Exercises

Chen Peng

Exercise 2.1 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?

Answer:

$$\mathbb{P}\left(A_t = \arg\max_a Q_t(a)\right) = 1 - \epsilon + \frac{\epsilon}{2} = 0.75. \tag{1}$$

Exercise 2.2: Bandit example 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?

Answer: To determine whether the ϵ case occurs, we should check whether the selected action corresponds to the highest action value on each time step, as presented in Table 1.

Time t	$Q_t(1)$	$Q_t(2)$	$Q_t(3)$	$Q_t(4)$	$\arg\max_a Q_t(a)$	A_t	ϵ case
1	0	0	0	0	1, 2, 3, 4	1	Possibly
2	$(-1)/1 = -1$	0	0	0	2, 3, 4	2	Possibly
3	$(-1)/1 = -1$	$1/1 = 1$	0	0	2	2	Possibly
4	$(-1)/1 = -1$	$(1-2)/2 = -1/2$	0	0	3, 4	2	Definitely
5	$(-1)/1 = -1$	$(1-2+2)/2 = 1/2$	0	0	2	3	Definitely

Table 1: Evolution of the action values as learning proceeds.

Exercise 2.3 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.

Answer: In the long run, both ϵ-greedy methods are able to identify the best action and the probability of selecting the best action can be calculated as

$$\mathbb{P}\left(A_t = \arg\max_a Q_t(a)\right) = 1 - \epsilon + \frac{\epsilon}{10} = 1 - 0.9\epsilon, \tag{2}$$

which is 91% for $\epsilon = 0.1$ and 99.1% for $\epsilon = 0.01$. The (expected) cumulative reward can be expressed as

$$\mathbb{E}\left[\sum_{t=1}^{T} R_t\right] = (1 - 0.9\epsilon) \cdot q_*(a_*) + (0.9\epsilon) \cdot 0 = (1 - 0.9\epsilon) q_*(a_*), \tag{3}$$

where $q_*(a_*)$ denotes the expected value of the optimal action. Thus in the long run, the ϵ-greedy method with $\epsilon = 0.01$ will perform best in terms of cumulative reward and probability of selecting the optimal action.

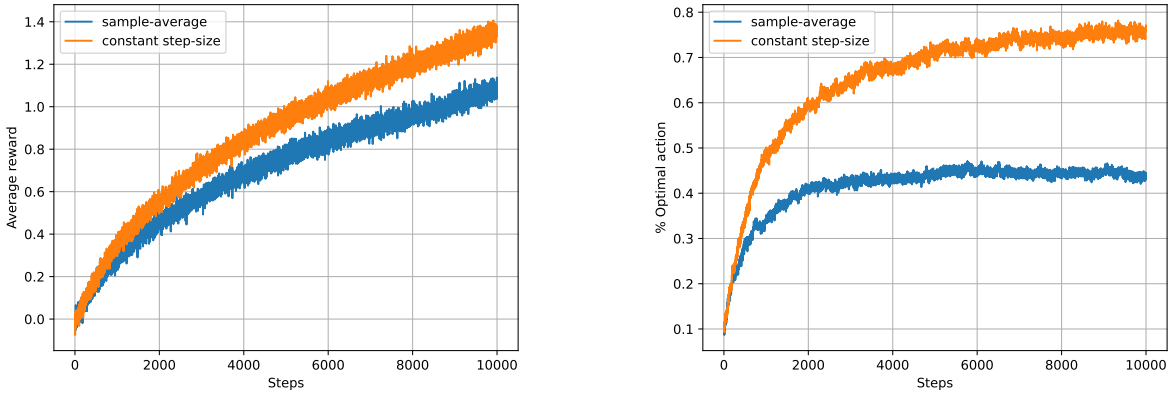
Exercise 2.4 If the step-size parameters, α_n, are not constant, then the estimate Q_n is a weighted average of previously received rewards with a weighting different from that given by (2.6). What is the weighting on each prior reward for the general case, analogous to (2.6), in terms of the sequence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

Answer: The weighting can be derived as follows.

$$
\begin{aligned}
Q_{n+1} &= \alpha_n R_n + (1-\alpha_n) Q_n && (4)\\
&= \alpha_n R_n + (1-\alpha_n)[\alpha_{n-1} R_{n-1} + (1-\alpha_{n-1}) Q_{n-1}] && (5)\\
&= \alpha_n R_n + (1-\alpha_n)\alpha_{n-1} R_{n-1} + (1-\alpha_n)(1-\alpha_{n-1}) Q_{n-1} && (6)\\
&= \sum_{i=1}^{n} \prod_{j=i+1}^{n} (1-\alpha_j) \cdot \alpha_i R_i + \prod_{j=1}^{n} (1-\alpha_j) \cdot Q_1. && (7)
\end{aligned}
$$

Exercise 2.5 (programming) 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 zero 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.

Answer: The results are shown in Fig. 1, which demonstrates that the sample-average method performs worse than the constant step-size method when applied to nonstationary bandits.



(a) Average reward

(b) Optimal action percentage

Figure 1: Average performance of ϵ-greedy action-value methods on the 10-armed testbed. These data are averages over 2000 runs with different bandit problems. The sample-average and constant step-size methods are considered with $\epsilon = 0.1$.

Exercise 2.6: Mysterious Spikes 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?

Answer: During the first 10 steps, actions are selected in turn due to the optimistic initial values. On step 11, since each action has been selected once, the action-value estimates are uniformly biased by the initial values. As a result, it is highly likely that the greedy action aligns with the optimal action. However, because the received reward on step 11 is still significantly lower than the estimate, the agent shifts to exploring other actions, leading to the first noticeable spike. Subsequently, as the different bandit tasks evolve asynchronously, the following spikes are less significant.

Exercise 2.7: Unbiased Constant-Step-Size Trick 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

$$\beta_n \doteq \alpha/\bar{o}_n, \tag{8}$$

to process the nth 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:

$$\bar{o}_n \doteq \bar{o}_{n-1} + \alpha(1 - \bar{o}_{n-1}), \quad \text{for } n \geq 0, \quad \text{with } \bar{o}_0 \doteq 0. \tag{9}$$

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

Answer: In this case, the estimate can be written as

$$Q_{n+1} = \beta_n R_n + (1 - \beta_n)Q_n = \sum_{i=1}^{n} \prod_{j=i+1}^{n} (1 - \beta_j) \cdot \beta_i R_i + \prod_{j=1}^{n} (1 - \beta_j) \cdot Q_1. \tag{10}$$

Further, we can rewrite the weight associated with Q_1 as

$$\prod_{j=1}^{n}(1 - \beta_j) = \prod_{j=1}^{n}\left(1 - \frac{\alpha}{\bar{o}_j}\right) = \left(1 - \frac{\alpha}{\bar{o}_1}\right) \cdot \prod_{j=2}^{n}\left(1 - \frac{\alpha}{\bar{o}_j}\right) = \left(1 - \frac{\alpha}{\alpha}\right) \cdot \prod_{j=2}^{n}\left(1 - \frac{\alpha}{\bar{o}_j}\right) = 0, \tag{11}$$

which confirms that Q_n does not suffer from initial bias.

Exercise 2.8: UCB Spikes 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.

Answer: During the first 10 steps, each action is selected once without preference, resulting in a relatively low average reward. On the 11th step, since the uncertainty measure is equal across all actions, the UCB algorithm selects an action based solely on estimated values. The action with the highest estimate is likely to yield a high reward at this point. Subsequently, as the uncertainty associated with the selected actions decreases, the agent may begin exploring other actions with lower estimated values, leading to a drop in rewards over the following steps. When c is reduced to 1, the influence of the uncertainty term diminishes, causing the agent to favor actions with higher estimates, which in turn leads to a less prominent spike.

Exercise 2.9 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.

Answer: In the case of two actions, the probability can be written as

$$\mathbb{P}(A_t = a) = \frac{e^{H_t(a)}}{e^{H_t(a)} + e^{H_t(b)}} = \frac{1}{1 + e^{-(H_t(a) - H_t(b))}} = \sigma(H_t(a) - H_t(b)), \tag{12}$$

which is the logistic function.

Exercise 2.10 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 0.1 and 0.2 with probability 0.5 (case A), and 0.9 and 0.8 with probability 0.5 (case B). If you are not able to tell which case you face at any step, what is the best expectation of success 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 expectation of success you can achieve in this task, and how should you behave to achieve it?

Answer: Note that the optimal action is 2 in Case A and 1 in Case B. When the current case is unknown, the probability of selecting the optimal action is at most 0.5, which can be achieved by choosing actions at random. However, when the current case is known, a bandit algorithm can be applied to learn the optimal action for each case individually. In this way, the probability of selecting the optimal action can approach 1.

Exercise 2.11 (programming) Make a figure analogous to Figure 2.6 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.

Answer: For the non-stationary case, the plot is provided in Fig. 2. As expected, the ϵ-greedy with constant step-size algorithm achieves the best performance, as it is designed to handle non-stationary bandits.

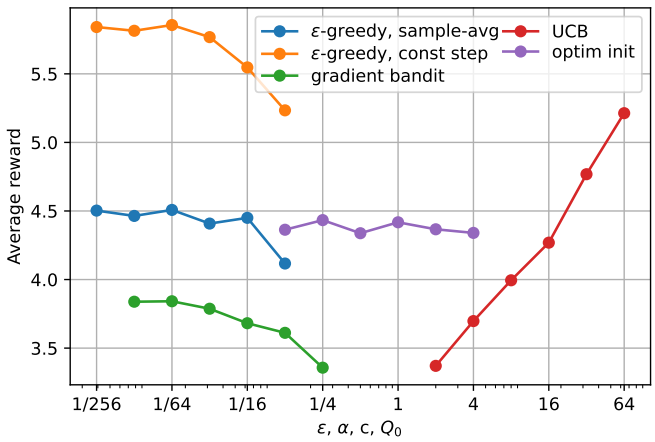


Figure 2: A parameter study of the various bandit algorithms. Each point is the average reward obtained over last $100,000$ steps with a particular algorithm at a particular setting of its parameter.