Chapter 2: Multi-armed Bandits

Nisheet Patel
Answers to textbook problems

August 22, 2019

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?

ϵ-greedy action selection takes place as follows:

$$A_t = \begin{cases} \arg\max_a Q_t(a) & \text{with probability} \quad 1-\epsilon \\ \text{random} & \text{with probability} \quad \epsilon \end{cases}$$

For two actions with $\epsilon = 0.5$, the greedy action is selected with probability 0.5 (greedy) + 0.5/2 (random) = 0.75

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 $Q_1(a) = 0\ \forall a$. 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?

The ϵ case definitely occurred on steps 2 & 5, and it may have occurred (but not necessarily) on steps 1, 3 & 4.

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.

In the long run, as the number of time-steps approaches infinity, the $\epsilon = 0.01$ will perform the best, selecting the best action 99% of the time and getting average reward 1.5445/action (1.55*0.99+1*0.01), whereas the $\epsilon = 0.1$ will select the optimal action only 90% of the time getting average reward 1.405/action (1.55*0.9+1*0.1).

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?

$$\begin{aligned}
Q_{n+1} &= Q_n + \alpha_n[R_n - Q_n] \\
&= \alpha_n R_n + (1-\alpha_n)Q_n \\
&= \alpha_n R_n + (1-\alpha_n)[\alpha_{n-1}R_{n-1} + (1-\alpha_{n-1})Q_{n-1}] \\
&= \alpha_n R_n + (1-\alpha_n)\alpha_{n-1}R_{n-1} + (1-\alpha_n)(1-\alpha_{n-1})\alpha_{n-2}R_{n-2} + ... + \prod_{i=1}^{n}(1-\alpha_i)Q_1
\end{aligned}$$

Thus, in general, the weighting on the reward k time-steps away is $\prod_{i=n-k+1}^{n}(1-\alpha_i)\alpha_{n-k}$.

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.

Skipped for now.

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?

With optimistic initial values, the greedy algorithm will explore for the first 10 steps (for the 10-armed bandit), after which the $Q_i(a) = \frac{<R|a>+5}{N(a)}$ for $i > 10$, i.e. it will be the sample average plus a bias term introduced by the optimistic initial value. Since the plot is over 2000 task runs, the first 10 samples will reveal the best option with high probability, leading to the spike in the 11th action, since the bias term for all 10 options is the same. Furthermore, since the bias of the optimistic initial value persists for a while, we expect subsequently smaller spikes in the 21st, 31st, 41st... actions.

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

$$\beta_n = \alpha/\bar{o}_n,$$

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:*

$$\bar{o}_n = \bar{o}_{n-1} + \alpha(1 - \bar{o}_{n-1}), \textit{for } n \geq 0, \textit{with } \bar{o}_0 = 0.$$

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

Let us first derive the form of β_n.

$$\begin{aligned}
\bar{o}_n &= \bar{o}_{n-1} + \alpha(1 - \bar{o}_{n-1}) \\
&= \alpha + (1-\alpha)\bar{o}_{n-1} \\
&= \alpha + (1-\alpha)[\alpha + (1-\alpha)\bar{o}_{n-2}] \\
&= \alpha + (1-\alpha)\alpha + (1-\alpha)^2\alpha + ... + (1-\alpha)^{n-1}\alpha + (1-\alpha)^n \cancelto{0}{\bar{o}_0}
\end{aligned}$$

Thus,

$$\begin{aligned}
\beta_n &= \frac{1}{\sum_{i=0}^{n-1}(1-\alpha)^i} \\
\beta_1 &= \frac{1}{1} \\
\beta_2 &= \frac{1}{1 + (1-\alpha)} \\
\beta_3 &= \frac{1}{1 + (1-\alpha) + (1-\alpha)^2}
\end{aligned}$$

Alternatively, we can also see that $\beta_1 = 1$ from $\bar{o}_1 = \alpha$, $\bar{o}_0 = 0$. Notice that $0 < (1-\alpha)^i < 1 \ \forall i$, and hence the denominator increases and β_n decreases with n. Thus, $0 < \beta_i < 1 \ \forall i$ and $\beta_i < \beta_j$ iff $i < j$. We can also formally show that in the continuous limit, $\beta(n) \propto exp(\frac{-(1-\alpha)^n}{n})$, and thus, it decays to zero as $\lim_{n\to\infty} \beta(n) = 0$.

From *Exercise 2.4*, we know that:

$$Q_{n+1} = \sum_k w_k R_k + \cancelto{0}{\prod_{i=1}^{n}(1-\beta_i)Q_1}$$

where $w_k = \prod_{i=n-k+1}^{n}(1-\beta_i)\beta_{n-k}$, and the second term disappears because the factor $(1-\beta_1) = 0$. Thus, the initial bias disappears and the weights w_k decay as well.

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.*

The UCB is forced to explore all options before exploiting since its criterion for selecting an action is infinity before it has chosen it. Since there are 10 options, on the 11th option, we expect a definitive spike because it now has one sample from each option, and is exploiting based on this. Once it does so, if the option is not that much better than some others, the criterion reduces because the arm has been visited twice, whereas every other arm has been visited only once, leading it to explore again. If the weight on the exploration term is less prominent, c=1 instead of 2, then the spike should be less prominent too.

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.

In the case of two actions, the probability of choosing option 1 given by the softmax distribution for preferences $H_t(1), H_t(2)$ for actions $1, 2$ at time t is:

$$\begin{aligned}\pi_t(1) &= \frac{e^{H_t(1)}}{e^{H_t(1)} + e^{H_t(2)}} \\ &= \frac{1}{1 + e^{H_t(2) - H_t(1)}} \\ &= sigmoid\big(H_t(1) - H_t(2)\big)\end{aligned}$$

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 dont 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?

If we're unable to tell which case we face, then it equivalent to having probability (0.9+0.1)/0.5 = 0.5 for action 1 and 0.5 for action 2. In this case, we can behave in a greedy manner, ϵ-greedy manner, use the UCB algorithm, or set optimistic initial values. Eventually, the probabilities will converge to choosing randomly and the best expectation of success is getting rewarded 0.5 units/action.

If we are able to tell which case we face, however, the best expected reward in case is 0.2 units/action in case A, and 0.9 units/action in case B, yielding the best expected reward of (0.9+0.2)*0.5 = 0.55/action. Here, we should keep separate action values for each case for both actions, and use and update the action values in a case-dependent manner. Any of the algorithms (except greedy) will eventually converge to the true reward probabilities and yield optimal behaviour, but for quick results, we may choose ϵ-greedy starting with a high ϵ and gradually lowering it, or the sample-average method, or UCB. Typically, UCB performs better, so we may choose to behave that way.