Reinforcement Learning S&B

Tabular Solution Methods

Exercises for chapter 2 : Multi-armed bandits

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 proposal. ϵ defines the probability to take any action independantly of it's action-value estimate (including the greedy one). So, if we call p the probability of taking the greedy action and n the number of possible action to chose from, then $p = (1 - \epsilon) + \epsilon * \frac{1}{n}$ hence for $\epsilon = 0.5$ and $n = 2$, $p = 0.75$. ■

2. 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 proposal. Let's answer the second question quickly, the ϵ case might have occured in any of the step. Now on the first question we simply have to compute the action-values for each step. Since they are updated one at a time, we only note the updated one at each time step knowing that $Q_1(a) = 0$ for all a. So we have :

   $$Q_2(1) = -1, Q_3(2) = 1, Q_4(2) = -0.5, Q_5(2) = \frac{1}{3}, Q_6(3) = 0$$

   Each time an action with non maximal action-value, namely at step 4 and 5, is selected then we must be in the ϵ case. ■

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 proposal. Since $Q_t(a) \longrightarrow q_\star(a)$, in the long run, the estimate of the action-value should be the same in both case. However, the probability of selecting a suboptimal action is much higher (ten times) for $\epsilon = 0.1$ than for $\epsilon = 0.01$. Hence the probability of selecting the right action will be respectively 0.91 and 0.991 making it 1.089 times more probable to select the best action in the second case. In terms of cumulative reward, let's say the best action has mean reward r and the other action have mean reward r' averaged on all other actions. Then at each time step, the reward for $\epsilon = 0.01$ would be $0.991r + 0.009r'$ while it would be $0.91r + 0.09r$ for $\epsilon = 0.1$. ■

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 proposal. From the update rule we have :

$$
\begin{aligned}
Q_{n+1} &= Q_n + \alpha_n(R_n - Q_n) \\
Q_{n+1} &= \alpha_n R_n + (1-\alpha_n)Q_n \\
Q_{n+1} &= \alpha_n R_n + (1-\alpha_n)(\alpha_{n-1}R_{n-1} + (1-\alpha_{n-1})Q_{n-1}) \\
Q_{n+1} &= \alpha_n R_n + (1-\alpha_n)\alpha_{n-1}R_{n-1} + (1-\alpha_n)(1-\alpha_{n-1})Q_{n-1} \\
&\dots \\
Q_{n+1} &= \sum_{i=1}^{n} \alpha_i R_i \prod_{j=i}^{n-1}(1-\alpha_{j+1}) + \prod_{i=1}^{n}(1-\alpha_i)Q_1
\end{aligned}
$$

■

5. 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.

Answer proposal. See code. ■