2 Multi-Armed Bandits

2.1

In ε-greedy action selection, for the case of two actions and $\varepsilon = 0.5$, what is the possibility that a non-greedy action is selected?

- With probability $(1 - \varepsilon)$, the greedy action is chosen outright. Then, with probability ε, one of the two actions is chosen at random (with equal chance), so the non-greedy action is chosen with probability $\frac{\varepsilon}{2} = 0.25$

2.2 Bandit Example

Consider a k-armed bandit problem wiht $k = 4$ 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 thee time steps the ε case may have occured, causing an action to be selected at random. On which time steps did this definitely occur? On which time steps could this possibly have occured?

- On timestep 2 and 5, the greedy actions were 1 and 2 respectively, but different actions were chosen. Here, the ε case definitely occured.
- The ε case may have occured on all other timesteps. Since the ε case can still select the greedy action, it is not possible to rule out that this happened.

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 answers quantitatively.

- Both the $\varepsilon = 0.1$ and $\varepsilon = 0.01$ cases will eventually converge to the true probabilities. Let $R*$ be the reward when choosing the optimal action every time, and L the expected loss when choosing a suboptimal reward. Then the average reward will converge to $R * -\varepsilon \cdot L$, meaning there will be a difference of $0.09L$ between the two cases.
- In the long run, both algorithms will be (nearly) certain about what the best reward is. The $\varepsilon = 0.1$ agent will select the optimal action in $1 - \varepsilon = 90\%$ of cases. The $\varepsilon = 0.01$ agent will select the optimal action in $1 - \varepsilon = 99\%$ of cases.
- By both metrics, the lower ε is superior in the long run. However, it takes much longer to converge.

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?

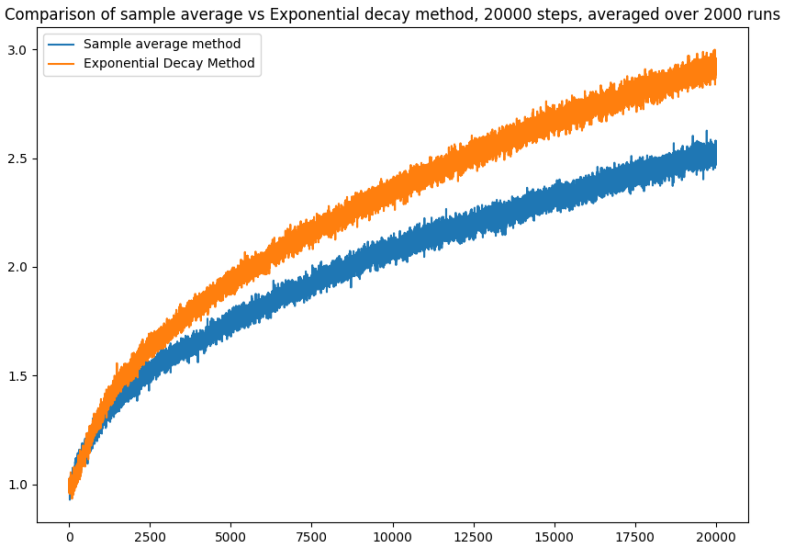
- This is fundamentally the same as in (2.6), only the notation comes out less cleanly because we cannot aggregate the different step size parameters into one neat exponent.

$$
\begin{aligned}
Q_{n+1} &= Q_n + \alpha_n[R_n + Q_n] && (1)\\
&= \alpha_n R_n + (1-\alpha_n)Q_n && (2)\\
&= \alpha_n R_n + (1-\alpha_n)[\alpha_{n-1}R_{n-1} + (1-\alpha_{n-1}Q_{n-1}] && (3)\\
&= \alpha_n R_n + (1-\alpha_n)\alpha_{n-1}R_{n-1} + (1-\alpha_n)(1-\alpha_{n-1})Q_{n-1} && (4)\\
&= \alpha_n R_n + (1-\alpha_n)\alpha_{n-1}R_{n-1} + (1-\alpha_n)(1-\alpha_{n-1}) && (5)\\
&\quad + ... + (1-\alpha_2)\alpha_1 R_1 && (6)\\
&= \sum_{i=1}^{n}\left[\alpha_i \cdot \left(\prod_{k=i+1}^{n}(1-\alpha_k)\right) R_i\right] && (7)
\end{aligned}
$$

2.5 programming

Design and conduct an experiment to demonstrate the difficulties that sample-average methods have for nostationary 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 wiht 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 $\varepsilon = 0.1$ and longer runs, say of 10.000 steps.

- For the code for this problem, see the Github. The plot below clearly shows that an exponential decay method is better equipped to deal with changing ground truths than a sample average method.



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?

- The initial value of 5 is very optimistic. In particular, it is very large relative to the step size. When the agent greedily chooses the optimal action based on an optimistic estimate, it will do so a few times in a row before the estimate has sufficiently converged to the true value.

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 in (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* 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.

$$\bar{o}_n = \bar{o}_{n-1} + \alpha(1 - \bar{o}_{n-1}), \text{ for } n \geq 0, \text{ with } \bar{o}_0 = 0 \tag{8}$$

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

We can easily see that

$$Q_{n+1} = Q_n + \beta_n(R_n - Q_n) \tag{9}$$
$$= (1 - \beta_n)Q_n + \beta_n R_n \tag{10}$$

Furhtermore,

$$\beta_1 = \alpha/\bar{o}_1 \tag{11}$$
$$= \alpha/(\bar{o}_0 + \alpha(1 - \bar{o}_0)), \text{ with } \bar{o}_0 = 0 \tag{12}$$
$$= \alpha/(0 + \alpha(1 - 0)) \tag{13}$$
$$= 1 \tag{14}$$

Now, substituting $\beta_1 = 1$ in 10 and specifying $n = 2$, we find that:

$$Q_2 = (1 - \beta_1)Q_1 + \beta_1 R_1 \tag{15}$$
$$= (1 - 1)Q_1 + R_1 \tag{16}$$
$$= R_1 \tag{17}$$

Which does not depend on our initial choice for Q, and is thus unbiased.

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.

- On the first 10 actions, the agent will cycle through the actions as a is considered a maximizing acition if $N(a)_t = 0$. Then, on step 11, it will then greedily choose an action for the first time, leading to a higher average reward. If c is large enough, it will then cycle through other actions again for a while, leading to lower average rewards.

2.9

Show that in the case of two actions, the soft-max distribution is te same as that given by the logistic, or sigmoid, funciton often used in statistics and artificial

neural networks.

$$P(A_t = a) = \frac{e^{H_t(a)}}{\sum_{b=1}^{k} e^{H_t(b)}}, \text{ for two actions:} \tag{18}$$

$$P(A_t = 1) = \frac{e^{H_t(1)}}{\sum_{b=1}^{2} e^{H_t(b)}} \tag{19}$$

$$= \frac{e^{H_t(1)}}{e^{H_t(1)} + e^{H_t(2)}} \tag{20}$$

$$= \frac{1}{1 + \frac{e^{H_t(2)}}{e^{H_t(1)}}} \tag{21}$$

$$= \frac{1}{1 + e^{H_t(2) - H_t(1)}}, \text{ substitute } \theta = H_t(1) - H_t(2) \tag{22}$$

$$= \frac{1}{1 + e^{-\theta}} \tag{23}$$

$$= \sigma(\theta) \tag{24}$$

2.10

Suppose you face a 2-armed bandit 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 succes 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?

- If you don't know which case you are presented with, you should simply track the reward for each action. The expected reward for both actions is the same since $0.5 \cdot 0.1 + 0.5 \cdot 0.9 = 0.5 \cdot 0.2 + 0.5 \cdot 0.8 = 0.5$. This is the best expectation you can achieve.
- If you do know which state you are presented with in advance, but you do not know the true values, you shoud track four rewards: one for each state-action combo. These should converge to the true probabilities fairly quickly, after which time you would select action 1 when you are presented with case A, and action 2 when you are presented with case B. The expectation of success you would achieve is $0.5 \cdot 0.9 + 0.5 \cdot 0.2 = 0.55$.

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 reward over the last 100,000 steps.

- Answer