Tags: #Exercises

Exercise 2.1

In ε-greedy action selection, for the case of two actions and ε = 0.5, what is the probability that the greedy action is selected?

Answer

0.75
0.5 chance that the greedy action would be selected outright, then 0.5 chance that we would explore. The exploratory action has two (2) possible actions with equal (uniform) probability to be selected, so the greedy action would have 0.5 probability of being selected in the exploratory action.

$$0.5 + (0.5 \times 0.5) = 0.75$$

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 $Q_1(a) = 0$, for all a. Suppose the initial sequence of actions and rewards is

$$\begin{aligned} A_1 &= 1, & R_1 &= -1 \\ A_2 &= 2, & R_2 &= 1 \\ A_3 &= 2, & R_3 &= -2 \\ A_4 &= 2, & R_4 &= 2 \\ A_5 &= 3, & R_5 &= 0 \end{aligned}$$

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

The ε case definitely occurred for A_4, because we see that A_2 gave a negative reward in the action prior, A_3. A_5 would also be an exploration step as well, since $Q(2) > Q(3)$ at this point, with $Q(2) = 1/3$ and $Q(3) = 0$.

The ε case could have possibly occurred at any time step.

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

The ε = 0.01 ε-greedy method is expected to perform the best in the very long run in terms of cumulative reward and probability of selecting the best action. ε = 0.01 exploits the best action 99% of the time, so it will converge closer to the optimal policy. The average reward would be higher than greedy or ε = 0.1 because it exploits the best action more often.

As time steps goes to infinity:
ε = 0.01 could potentially reach up to 99.1% of the optimal action.
ε = 0.1 would approach 91% of optimal actions.
ε = 0 (greedy) stays at about 35% of optimal actions.

So then,
For ε = 0.01:
$\text{Expected reward} = (0.991 \times 1.55) + (0.09 \times \text{average of suboptimal actions}) \approx 1.53605 + (0.09 \times \text{lower}$
For ε = 0.1:
$\text{Expected reward} = (0.91 \times 1.55) + (0.9 \times \text{average of suboptimal actions}) \approx 1.4105 + (0.9 \times \text{lower value}$
For ε = 0:
$\text{Expected reward} = (\sim 0.35 \times 1.55) \approx 0.5425$

For optimal action selection and average reward (ignoring the average of suboptimal actions):

- ε = 0.01 would be better than ε = 0.1 by about 8.9% (99.1% vs 91%, 1.53605 vs 1.4105)
- ε = 0.01 would be better than ε = 0 by about 183.1% (99.1% vs 35%, 1.53605 vs 0.5425)

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

So,

$$\begin{aligned} Q_{t+1} &= Q_t + \alpha_t[R_t - Q_t] \\ &= \alpha_t R_t + (1-\alpha_t)Q_t \end{aligned}$$

Expanding Q_t in terms of Q_{t-1} and each Q_{t-n} afterwards,

$$\begin{aligned} &= \alpha_t R_t + (1-\alpha_t)\alpha_{t-1}R_{t-1} + (1-\alpha_t)(1-\alpha_{t-1})Q_{t-1} \\ &= \alpha_t R_t + (1-\alpha_t)\alpha_{t-1}R_{t-1} + (1-\alpha_t)(1-\alpha_{t-1})\alpha_{t-2}R_{t-2} + (1-\alpha_t)(1-\alpha_{t-1})(1-\alpha_{t-2})Q_{t-2} \\ &= \alpha_t R_t + \alpha_{t-1}(1-\alpha_t)R_{t-1} + \alpha_{t-2}(1-\alpha_t)(1-\alpha_{t-1})R_{t-2} \\ &\quad + \cdots + \alpha_1(1-\alpha_t)(1-\alpha_{t-1})\cdots(1-\alpha_2)R_1 + \left(\prod_{i=1}^{t}(1-\alpha_i)\right)Q_1 \end{aligned}$$

And more generally,

$$= \sum_{k=1}^{t}\left(\alpha_k \prod_{i=k+1}^{t}(1-\alpha_i)\right)R_k + \left(\prod_{i=1}^{t}(1-\alpha_i)\right)Q_1$$

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 $\varepsilon = 0.1$ and longer runs, say of 10,000 steps.

Answer

OMITTED PROGRAMMING EXERCISE

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

The optimistic curve is by greedy selection of actions. Initially, since we are initializing it with optimistic initial values, it will be encouraged to explore every action randomly. Even the optimal action's value will be decreased once it has been selected at first until the value is updated toward the actual value eventually. Non-optimal actions are updated to lower values faster than the optimal action. Because of all the exploration and the model being "disappointed" until values trend toward their actual values, this method performs worse on average on early steps.

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:

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

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

Answer

By expanding and substituting, we see that:

$$
\begin{aligned}
Q_{n+1} &= Q_n + \beta_n \left[R_n - Q_n\right] \\
&= \beta_n R_n + (1-\beta_n) Q_n \\
&= \frac{\alpha}{\bar{o}_n} R_n + (1 - \frac{\alpha}{\bar{o}_n}) Q_n \\
&= \frac{\alpha}{\bar{o}_n} R_n + (1 - \frac{\alpha}{\bar{o}_n}) [\frac{\alpha}{\bar{o}_{n-1}} R_{n-1} + (1 - \frac{\alpha}{\bar{o}_{n-1}}) Q_{n-1}] \\
&= \frac{\alpha}{\bar{o}_n} R_n + (1 - \frac{\alpha}{\bar{o}_n}) \frac{\alpha}{\bar{o}_{n-1}} R_{n-1} + (1 - \frac{\alpha}{\bar{o}_n})(1 - \frac{\alpha}{\bar{o}_{n-1}}) Q_{n-1} \\
&= \frac{\alpha}{\bar{o}_n} R_n + (1 - \frac{\alpha}{\bar{o}_n}) \frac{\alpha}{\bar{o}_{n-1}} R_{n-1} + (1 - \frac{\alpha}{\bar{o}_n})(1 - \frac{\alpha}{\bar{o}_{n-1}}) \frac{\alpha}{\bar{o}_{n-2}} R_{n-2} + \cdots \\
&\quad + (1 - \frac{\alpha}{\bar{o}_n})(1 - \frac{\alpha}{\bar{o}_{n-1}}) \cdots (1 - \frac{\alpha}{\bar{o}_1}) Q_1
\end{aligned}
$$

Simplify the coefficients with $\bar{o}_n = \bar{o}_{n-1} + \alpha(1 - \bar{o}_{n-1})$. We can rewrite this as:

$$
\begin{aligned}
\bar{o}_n &= \bar{o}_{n-1} + \alpha - \alpha \bar{o}_{n-1} \\
&= (1-\alpha)\bar{o}_{n-1} + \alpha
\end{aligned}
$$

Using this, we can simplify the coefficient of R_{n-k}:

$$
\begin{aligned}
(1 - \frac{\alpha}{\bar{o}_n})(1 - \frac{\alpha}{\bar{o}_{n-1}}) \cdots (1 - \frac{\alpha}{\bar{o}_{n-k+1}}) \frac{\alpha}{\bar{o}_{n-k}} &= \frac{(\bar{o}_n - \alpha)(\bar{o}_{n-1} - \alpha) \cdots (\bar{o}_{n-k+1} - \alpha)\alpha}{\bar{o}_n \bar{o}_{n-1} \cdots \bar{o}_{n-k}} \\
&= \frac{((1-\alpha)\bar{o}_{n-1})((1-\alpha)\bar{o}_{n-2}) \cdots ((1-\alpha)\bar{o}_{n-k})\alpha}{\bar{o}_n \bar{o}_{n-1} \cdots \bar{o}_{n-k}} \\
&= \frac{(1-\alpha)^k \alpha}{\bar{o}_n}
\end{aligned}
$$

This shows exponential decay of $(1-\alpha)^k$ as we go back in time (k increases). This is scaled by α and normalized by $\bar{o}_n$. So more recent rewards have higher coefficients, giving them more weight.

For the coefficient of Q_1,

$$
(1 - \frac{\alpha}{\bar{o}_n})(1 - \frac{\alpha}{\bar{o}_{n-1}}) \cdots (1 - \frac{\alpha}{\bar{o}_1}) = \prod_{i=1}^{n} \left(1 - \frac{\alpha}{\bar{o}_i}\right)
$$

Now,

$$
\lim_{n \to \infty} \prod_{i=1}^{n} \left(1 - \frac{\alpha}{\bar{o}_i}\right) = 0
$$

which means the coefficient of Q_1 approaches 0 as n increases. Consequently, the influence of

the initial condition Q_1 on Q_n diminishes to zero over time. Q_n becomes dominated by reward terms ($R_n, R_{n+1}, etc.$) as n increases.

Therefore, the initial condition decays exponentially and approaches zero, meaning there is no initial bias. The estimation becomes more and more dependent on more recent rewards. It effectively "forgets" the initial condition over time and will adapt to the most recent data.

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

Once you try all 10 actions once (like in the first 10 steps), the 11th action selects the most promising arm based on the observed rewards and the uncertainty. This action generally has a high reward because it is better informed about the reward of each action, causing the spike. After this 11th step, UCB explores other arms that may be suboptimal, which decreases the average reward on subsequent steps.

$c = 1$ would have a less prominent spike because the exploration bonus is lower, which means there is relatively more weight on the exploitation term $Q_t(a)$ and it would explore less, meaning that the decrease in average reward due to exploring would be less pronounced.

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.

Soft-max for K actions:

$$\sigma(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

Sigmoid for two (2) actions:

$$\sigma(x) = \frac{e^x}{1 + e^x}$$

Soft-max for two (2) actions (focusing on one action):

$$\sigma(\mathbf{z})_1 = \frac{e^{z_1}}{e^{z_1} + e^{z_2}}$$

Now if we write soft-max in terms of $x = z_1 - z_2$, defining x as the relative difference between inputs:

$$\begin{aligned}
\sigma(\mathbf{z})_1 &= \frac{e^{z_1}}{e^{z_1} + e^{z_1 - x}} \\
&= \frac{1}{1 + e^{-x}} \\
&= \frac{e^x}{1 + e^x} \\
&= \sigma(x)
\end{aligned}$$

And same holds true for $\sigma(\mathbf{z})_2$ with $x = z_2 - z_1$.

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

Case Unknown

$$\begin{aligned}
\mathbb{E}[\text{Action 1}] &= 0.5 \times 0.1 + 0.5 \times 0.9 = 0.5 \\
\mathbb{E}[\text{Action 2}] &= 0.5 \times 0.2 + 0.5 \times 0.8 = 0.5
\end{aligned}$$

Both actions have the same expected value. The best expectation of success is 0.5.

To behave optimally,

- Just pick an action randomly, as they both have the same expectation of success.

Case Known

Best expectation of success:

- For case A: Choose action 2 (value 0.2)
- For case B: Choose action 1 (value 0.9)
  Expected success:

$$\mathbb{E}[\text{success}] = 0.5 \times 0.2 + 0.5 \times 0.9 = 0.55$$

  To behave optimally,
- If case A: Choose action 2.
- If case B: Choose action 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.

OMITTED PROGRAMMING EXERCISE