RL Excercise Chapter 2

huseyinabanox

June 2022

2 Exercises

2.1 Question

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

Greedy action is selected with a probability of 0.5. Greedy action can also be selected in random action selection with a probability of $(1-\epsilon)*0.5 = 0.5*0.5 = 0.25$ Overal probability is $0.5 + 0.25 = 0.75$

2.2 Question

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 $\epsilon-$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

Random action selection may result in the selection of greedy action. Thus all steps may involve random action selection.

t	Q(A1)	Q(A2)	Q(A3)	Q(A4)	Greedy	Selected	Reward	Random?
1	0	0	0	0	Any	1	-1	Possible
2	-1	0	0	0	2,3,4	2	1	Possible
3	-1	1	0	0	2	2	-2	Unlikely
4	-1	-1/2	0	0	3,4	2	2	Yes
5	-1	1/3	0	0	2	3	0	Yes

If greedy action and selected actions are different then random selection is certain.

2.3 Question

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 optimal action a* is found. Even after finding optimal action, $\epsilon-$greedy method will continue exploring.

Optimal action selection:

Optimal action is selected with probability $1 - \epsilon$. Non-optimal action is selected with probability ϵ. Optimal action a* may be selected randomly with a probability ϵ/k. k is given as 10.

Total probability of selecting the optimal action is:

$$P(A_i = a^*) = 1 - \epsilon + \epsilon/10 \tag{1}$$

Considering given ϵ values:

$$\frac{P(A_i = a^*|\epsilon = 0.01)}{P(A_i = a^*|\epsilon = 0.1)} = \frac{1 - 0.01 + 0.01/10}{1 - 0.1 + 0.1/10} = 0.991/0.91 = 1.09 \tag{2}$$

Average reward:

Since they are derived from the same distributed with a mean of 0, expected value of all actions combined is 0. Thus average reward for random actions will be zero. We will only consider average reward for greedy action selections.

In the long run average reward will be equal to expected value of selecting greedy action:

$$E[R] = q^*(a^*) * (1 - \epsilon) \tag{3}$$

Considering given ϵ values:

$$\frac{E[R|\epsilon = 0.01]}{E[R|\epsilon = 0.1]} = \frac{q^*(a^*) * (1 - 0.01)}{q^*(a^*) * (1 - 0.1)} = \frac{0.99}{0.9} = 1.1 \tag{4}$$

2.4 Question

f 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

$$Q_{n+1} = Q_n + \alpha_n[R_n - Q_n] = \alpha_n R_n + (1-\alpha_n)Q_n \tag{5}$$

$$Q_{n+1} = \alpha_n R_n + (1-\alpha_n)[Q_{n-1} + \alpha_{n-1}[R_{n-1} - Q_{n-1}]] = \alpha_n R_n + (1-\alpha_n)[\alpha_{n-1}R_{n-1} + (1-\alpha_{n-1})Q_{n-1}] \tag{6}$$

$$Q_{n+1} = \alpha_n R_n + (1-\alpha_n)\alpha_{n-1}R_{n-1} + (1-\alpha_n)(1-\alpha_{n-1})Q_{n-1} \tag{7}$$

$$Q_{n+1} = \alpha_n R_n + (1-\alpha_n)\alpha_{n-1}R_{n-1} + (1-\alpha_n)(1-\alpha_{n-1})[\alpha_{n-2}R_{n-2} + (1-\alpha_{n-2})Q_{n-2}] \tag{8}$$

$$Q_{n+1} = \alpha_n R_n + (1-\alpha_n)\alpha_{n-1}R_{n-1} + (1-\alpha_n)(1-\alpha_{n-1})\alpha_{n-2}R_{n-2} + (1-\alpha_n)(1-\alpha_{n-1})(1-\alpha_{n-2})Q_{n-2} \tag{9}$$

$$Q_{n+1} = \sum_{i=0}^{n-1} \alpha_{n-i}R_{n-i} \prod_{k=0}^{i-1}(1-\alpha_{n-k}) + Q_1 \prod_{i=0}^{n-1}(1-\alpha_{n-i}) \tag{10}$$

2.5 Question

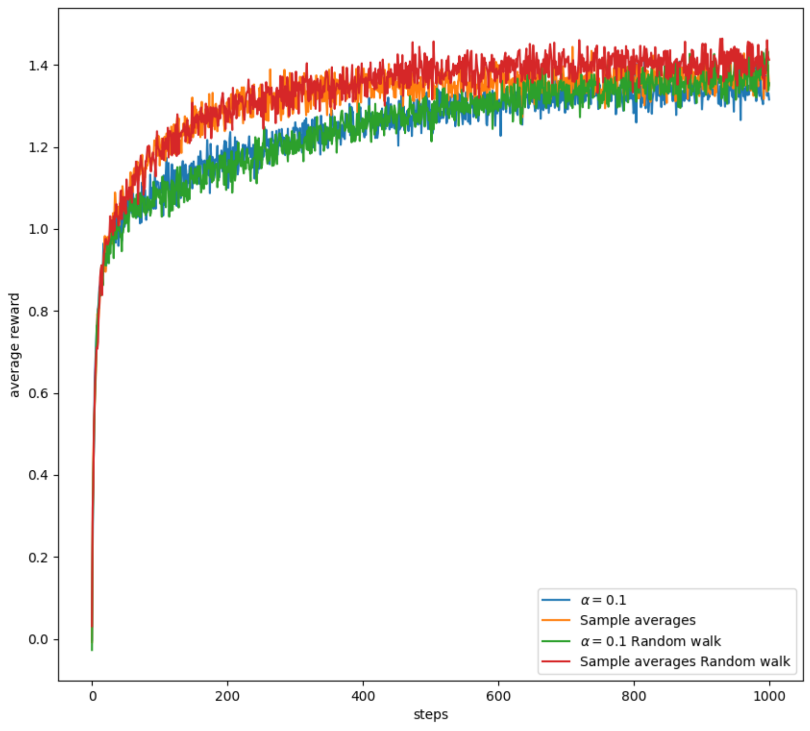
In the comparison shown in Figure 2.2, which method will perform best in the long run in terms of cumulative reward and probability of selecting the best action? How much better will it be? Express your answer quantitatively.

Answer

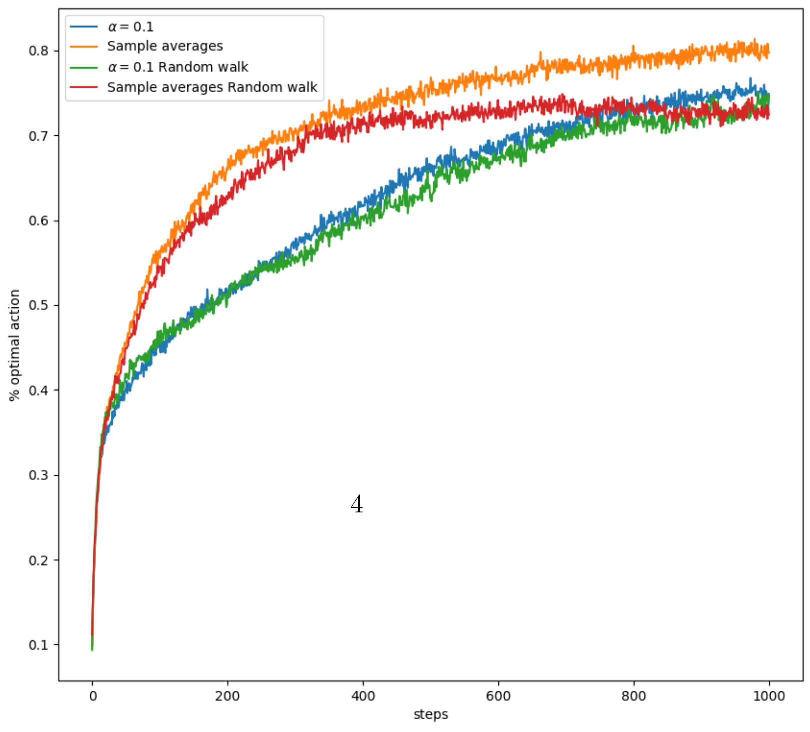
Aqui hice una correccion muy seria

First two bandits do not make random walk. Last two bandits make random walk. Constant step size performs better in case of random walk which creates a non-stationary target.

Well honestly I was expecting to see a remarkable difference between the effectiveness of the constant step-size value estimator vs sample-average when upgrading to non-stationary random walks. However, at least when the timeline is of 1000 steps, the is no big difference.



average reward
steps
$\alpha = 0.1$
Sample averages
$\alpha = 0.1$ Random walk
Sample averages Random walk



% optimal action
steps
$\alpha = 0.1$
Sample averages
$\alpha = 0.1$ Random walk
Sample averages Random walk

2.6 Question

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

Figure 2.3 shows optimal action selection behaviour. Optimistic initial value encourages exploration. Since all arms have equally optimistic high values, arms are selected randomly. Once the optimum action is selected its value is downgraded through the true value. In following few steps, non-optimal actions with higher value are selected. Non-optimal actions are downgraded faster than the optimal action.

Optimistic strategy explores early and learns obtains a good view about the optimum behaviour. Even though it oscilates dramatically in the beginning it improves faster. Optimistic method stops exloring once actual values are learnt which is good only for stationary problems.

Realistic method with $\epsilon-$greedy exploration explorer less frequent in the beginning thus learns more slowly.

For non-stationary problems $\epsilon-$greedy performs better since it continues exploration even after initial stage.

2.7 Question

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

Answer

Hice una correccion muy seria. No entendí muy bien la matemática en detalle del por qué el peso del $i-$ésimo reward se va haciendo más pequeño a medida que n avanza, pero lo cierto es que por lo menos tiene sentido la argumentación. Saqué esta solucion del primer solucionario que miré.

First consider this alternative response to exercise 2.4.

Let $\alpha_0 = 1$, then

$$Q_{n+1} = \left(\prod_{i=1}^{n}(1-\alpha_i)\right)Q_1 + \sum_{i=1}^{n}\alpha_i R_i \prod_{k=i+1}^{n}(1-\alpha_k). \tag{11}$$

Where $\prod_{i=x}^{y} f(i) \doteq 1$ if $x > y$.

Now we go for question 2.7. Consider the alternativ answer to Exercise 2.4. There is no dependence of Q_k on Q_1 for $k > 1$ since $\beta_1 = 1$. Now it remains to show that the weights in the remaining sum decrease as we look further into the past. That is

$$w_i = \beta_i \prod_{k=i+1}^{n} (1 - \beta_k) \tag{12}$$

increases with i for fixed n. For this, observe that

$$\frac{w_{i+1}}{w_i} = \frac{\beta_{i+1}}{\beta_i(1 - \beta_{i+1})} = \frac{1}{1 - \alpha} > 1 \tag{13}$$

where we have assumed $\alpha < 1$. If $\alpha = 1$ then $\beta_t = 1 \; \forall \, t$.

2.8 Question

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

In the first ten steps it is just randomly exploring (forced to explore), so the reward is low. Then, in the eleventh step, it already has an idea on the upper bounds, and all the actions have the same uncertainty, so the agent chooses the optimal action. Then, **since UCB prioritizes higher uncertainty**, especially in the first steps, the agent is forced to keep exploring not optimal actions. Eventually, uncertainty is less and less important, and also smaller since actions have already been chosen multiple times. This translates to a pseudo-policy[1] where the priority is the estimation over uncertainty, thus, taking better choices.

2.9 Question

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

$$Pr(A_1 = a) = \frac{e^{H_1(a)}}{e^{H_1(a)} + e^{H_1(b)}} = \frac{1}{1 + e^{H_1(b) - H_1(a)}} = sigmoid(H_1(a) - H_1(b)) \tag{14}$$

[1]I call it a pseudo policy because it is actually simpler that a policy. An actual policy is used when the state is not constant, and I am not talking about the case when the action-value is a random walk. I am talking about the case when, for example, the agent is facing multiple bandits but only one at a time, without actually knowing which bandit (with its own action-value distribution, which may be itself a random walk or not) he is facing.

2.10 Question

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 when we don't know the bandit we are facing:

In this case, the best thing we can do is to learn which is the value of each action itself (independently of the bandit). That is, learning $\mathbb{E}[R_i|A_i = 1] = \mathbb{E}[R_i|S_i = A, A_i = 1] * \mathbb{P}[S_i = A] + \mathbb{E}[R_i|S_i = B] * \mathbb{P}[S_i = B, A_i = 1]$, $\mathbb{E}[R_i|A_i = 2]$, and then choosing greedily the action with the best expectation, using whatever method that may fit the situation.

In this particular case, we have that $\mathbb{E}[R_i|A_i = 1] = \mathbb{E}[R_i|A_i = 2] = 0.5$. This means that in this situation our best expectation for the agent would be that it learns that it should choose uniformly action 1 and 2 [2]

Case where we do know the bandit we are facing:

In this case the agent has all the information it should need to learn an optimal policy. So we basically expect it to learn to choose greedily action 2 in case A, and action 2 in case B. Regarding the strategy (learning method), I would say a good option would be to basically run two independent learning agents, one for each bandit.

2.11 Question

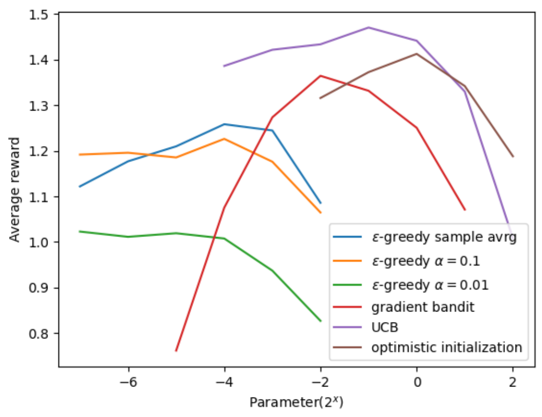
(programming) Make a figure analogous to Figure 2.6 for the nonstationary case outlined in Exercise 2.5. Include the constant-step-size "$\epsilon-$greedy algorithm with $\alpha = 0.1$. Use runs of 200,000 steps and, as a performance measure for each algorithm and parameter setting, use the average reward over the last 100,000 steps.

Answer

Below is the demanded graph. 200000 time steps was definitely taking extremely long, so we remained with 1000 time steps, and as a measure of effective for each parameter value, we used the mean value of the last 500 average rewards. One

[2] Although it may not be actually optimal, it is what we expect the agent to learn according to our requirements when choosing the learning method.

possible good option for future experiments is to try the 200000 time steps and, since for example in this case there were 36 bandits being tested with different strategies and parameter values, execute their graph in parallel.


Average reward
Parameter(2^x)
ε-greedy sample avrg
ε-greedy $\alpha = 0.1$
ε-greedy $\alpha = 0.01$
gradient bandit
UCB
optimistic initialization