RBE595 - Midterm

Keith Chester

Due date: March 26, 2023

Problem 1

In this question we are presented with two random variables with provided distributions $p = 0.2, 0.2, 0.2, 0.2, 0.1, 0.1$ and $q = 0.75, 0.0625, 0.0625, 0.0625, 0.0625$.

1.1

Calculate the entropy for each variable

To do this, we use the equation:

$$H(p) = -\sum_{i=1}^{n} p_i \log_2 p_i \tag{1}$$

...where n is the number of values within the distribution. Thus we can

$$H(p) = -\sum_{i=1}^{6} p_i \log p_i \ = -(0.2\log_2 0.2 + 0.2\log_2 0.2 + 0.2\log_2 0.2 + 0.2\log_2 0.2 + 0.1\log_2 0.1 + 0.1\log_2 0.1) \tag{2}$$

$$= -(4 * (0.2 * -2.32193) + 2 * (0.1 * -3.32193)) = 2.52193 \tag{3}$$

...and then for the next distribution:

$$H(q) = -\sum_{i=1}^{5} q_i \log_2 q_i \ = -(0.75\log_2 0.75 + 0.0625\log_2 0.0625 + 0.0625\log_2 0.0625 + 0.0625\log_2 0.0625 + 0.0625\log_2 0.0625) \tag{4}$$

$$= -((0.75 * -0.415) + 4 * (0.0625 * -4)) = 1.31125 \tag{5}$$

...thus p has 2.522 bits of entropy, and q has 1.311 bits of entropy.

1.2

Intuitively how can you tell which variable has a higher entropy without calculating the entropy numerically? What does higher entropy mean?

A random variable with high entropy means that the distribution/outcome is more uncertain/unpredictable. It's probability function is more spread out or has a wider ranger of possible values.Since q has single value it is skewed towards, we would suspect that p would have greater entropy; our calculations confirm this.

Problem 2

Which equation correctly relates $v*$ to $q*$? Draw the corresponding backup-diagram and explain.

- **A** - $v_*(s) = \sum_{r,s} \pi(a|s)p(s', r|s, a)[r + \gamma q_*(s')]$
- **B** - $v_*(s) = max_a q_*(s|a)$

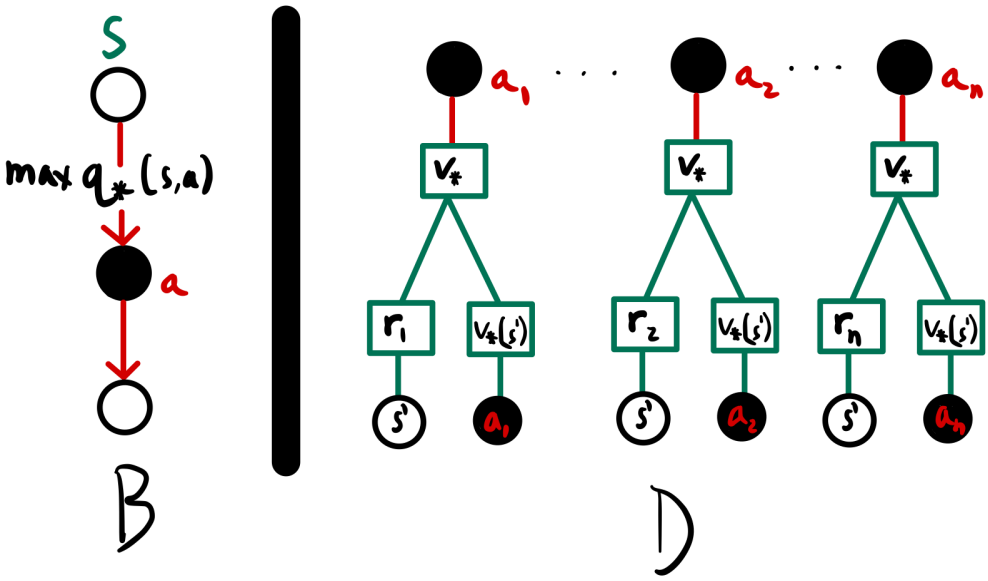

Figure 1: Backup diagram for B and D

- **C** - $v_*(s) = max_a \sum_{r,s} p(s', r|s, a)[r + \gamma q_*(s', a)]$
- **D** - $v_*(s) = max_a \sum_{r,s'} p(s', r|s, a)[r + \gamma v_*(s')]$
- **E** - None of the above
- **F** - B and D

Answer:
F - both **B** and **D**
You can see the backup diagrams for each drawn in Figure 1.
Reasoning:

- **A** - This is incorrect as it uses q_* in place of v_* in the backup equation.
- **B** - This directly computes the optimal state value via its greedy action of calculating the maximum of all actions and their resulting action values
- **C** - This is incorrect as it computes the expected value of the next state using the current state/action pair - it should use the next state/action pair s', a
- **D** - The definition of $q_*(s, a)$ as taking the expected return of state s with action a and taking the optimal thereafter; therefore we can take the maximum of all actions **D** returns and get the Bellman Equation.
- **E** -Since B and D is right, this is eliminated
- **F** - B and D are correct, so we choose this.

Problem 3

Consider a vehicle with 4 actions (left,right,up, down). There's no uncertainty in the outcome ofthe action (i.e. when left is commanded, the left state is achived). The actions that cause the vehicle outside the grid, leave the state unchanged. The reward for all transition is -1 except when the goal is reached where the reward is zero. Discount factor $\gamma = 1$

Figure 2 shows the name of the states and figure on the right shows the state-value $V(s)$*, for each state under a uniform random policy*

Termination	a	b	c
d	e	f	g
h	i	j	k
l	m	n	Termination

Termination	-14	-20	-22
-14	-18	?	-20
-20	-20	-18	-14
-22	-20	-14	Termination

Figure 2: Problem 3: States and $V(s)$

3.1

What is $q(k, \downarrow)$?

$$q(k, \downarrow) = r + \gamma V(k) = -1 + 1 * -14 = -15 \tag{6}$$

3.2

What is $q(g, \downarrow)$?

$$q(k, \downarrow) = r + \gamma V(g) = -1 + 1 * -20 = -21 \tag{7}$$

3.3

What is $V(f)$?

...where $p(s)$ is the probability of choosing a given state s', which for our initial calculation we will use uniform probability or $\frac{1}{4}$.

$$V(s) = \frac{1}{4}(-1 + \gamma * V(b)) + \frac{1}{4}(-1 + \gamma * V(e)) + \frac{1}{4}(-1 + \gamma * V(g)) + \frac{1}{4}(-1 + \gamma * V(j)) \tag{8}$$

$$V(s) = \frac{1}{4}(-1 + 1 * -20) + \frac{1}{4}(-1 + 1 * -18) + \frac{1}{4}(-1 + 1 * -20) + \frac{1}{4}(-1 + 1 * -18) \tag{9}$$

$$V(s) = \frac{-21}{4} + \frac{-19}{4} + \frac{-21}{4} + \frac{-19}{4} = \frac{-80}{4} = -20 \tag{10}$$

Problem 4

Consider the state-action interaction in Figure 3. The $q(s, a)$ written next to each action (left and right) is the initial estimate.

Consider a discount factor $\gamma = 0.5$ and learning rate $\alpha = 0.1$, and a reward transition of $r = 1$

4.1

What is the target value as well as updated value of $q(s, \leftarrow)$ using SARSA algorithm if the action at s' was the left action.

The update rule:

$$q(s, a) \leftarrow q(s, a) + \alpha[R + \gamma q(s_{t+1}, a) - Q(s, a)] \tag{11}$$

...and the target value is:

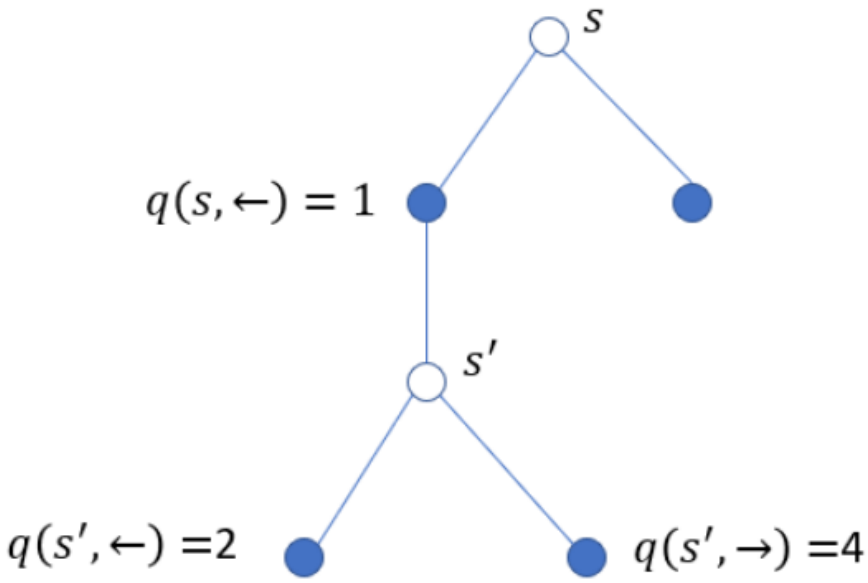

Figure 3: Problem 4: State-Interaction Tree

$$R_{t+1} + \gamma Q(s_{t+1}, a) \tag{12}$$

So first we calculate target value:

$$= 1 + 0.5 * 2 = 2 \tag{13}$$

...and then applying for the update value:

$$q(s, a) \leftarrow 1 + 0.1 * (2 - 1) = 1.1 \tag{14}$$

4.2

What is the target value as well as updated value of $q(s, \leftarrow)$ using SARSA algorithm if the action at s' was the right action.

Target value:

$$= 1 + 0.5 * 4 = 3 \tag{15}$$

Update value:

$$q(s, a) \leftarrow 1 + 0.1 * (3 - 1) = 1.2 \tag{16}$$

4.3

Assume that the action at s' has a distribution in such a way that it is 30 percent left action and 70 percent right action. What is the expected SARSA target value and as well expected value of $q(s, \leftarrow)$ under SARSA algorithm?

The target value of Expected SARSA is:

$$R_{t+1} + \gamma \sum_{a} \pi(a|s_{t+1}) q(s_{t+1}, a) \tag{17}$$

Target value:

$$= 1 + 0.5 * (0.30 * 2 + 0.70 + 4) = 2.7 \tag{18}$$

Update value:

$$q(s, a) \leftarrow 1 + 0.1 * (2.7 - 1) = 1.17 \tag{19}$$

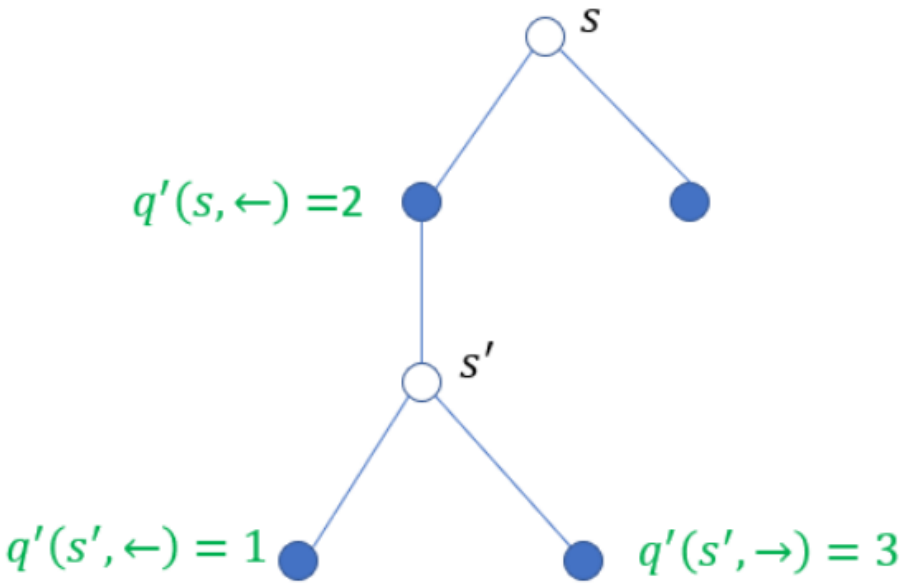

Figure 4: Problem 4: State-Interaction Tree Part 2

4.4

What is the target value as well as updated value of $q(s, \leftarrow)$ using Q-learning algorithm.

Our target value formula is:

$$R_{t+1} + \gamma max_a q(s_{t+1}, a) \tag{20}$$

...and our update value is:

$$q(s,a) \leftarrow q(s,a) + \alpha \left[r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s,a) \right] \tag{21}$$

So our target value is:

$$1 + 0.5 * 4 = 3 \tag{22}$$

...and our update value is:

$$q(s,a) \leftarrow 1 + 0.1 * (3 - 1) = 1.2 \tag{23}$$

4.5

Does the distribution of the action at s' have an effect on the Q-learning target value?

Yes, it does have an effect. The Q-learning target involves taking max of an action value over all possible states for s', which in turn depends upon the distribution/probabilities of choosing actions in that state.

4.6

Consider now we have another batch of initial estimates for the action-values. We call then $q'(s, a)$. They are shown in the diagram in Figure 4 in green color.

What is the updated value of $q(s, \leftarrow)$ using double-Q learning algorithm? (You need to use both green and black initial estimates)

$$q_1(s,a) \leftarrow q_1(s,a) + \alpha \left[r_{t+1} + \gamma q_2(s_{t+1}, \arg\max_a q_1(s_{t+1}, a)) - q_1(s,a) \right] \tag{24}$$

$$q_2(s,a) \leftarrow q_2(s,a) + \alpha\big[r_{t+1} + \gamma q_1(s_{t+1}, \arg\max_a q_2(s_{t+1},a)) - q_2(s,a)\big] \tag{25}$$

So for this problem we look at $q_1(s, \leftarrow)$:

$$q_1(s,a) \leftarrow 1 + 0.1 * (1 + (0.5 * 3) - 1) = 1.15 \tag{26}$$

4.7

What is the updated value of $q'(s, \leftarrow)$ using double-Q learning algorithm? (You need to use both green and black initial estimates)

For $q'(s, \leftarrow)$ we use the $q_2(s, \leftarrow)$ calculation:

$$q_2(s'a) \leftarrow 2 + 0.1 * (1 + (0.5 * 4) - 2) = 2.1 \tag{27}$$

Problem 5

As we discussed, n-step off-policy return via bootstrapping can be written as:

$$G_{t:h} = p_t(R_{t+1} + \gamma G_{t+1:h}) \tag{28}$$

where $p = \frac{\pi(A_t|S_t)}{b(A_t|S_t)}$ is the importance sampling ratio between target and behavior policy.
Using control variates, the return can be written as:

$$G_{t:h} = p_t(R_{t+1} + \gamma G_{t+1:h}) + (1 - p_t)V_{h-1}(S_t) \tag{29}$$

Prove that introducing the control variates in equation (2) does not add any bias to the original return (i.e. equation (1)) in expectation.

We start with our original equation as instructed...

$$G_{t:h} = p_t(R_{t+1} + \gamma G_{t+1:h}) \tag{30}$$

...then add in the control variate.

$$G_{t:h} = p_t(R_{t+1} + \gamma G_{t+1:h}) + (1 - p_t)V_{h-1}(S_t) \tag{31}$$

We take the expected value of the probability of each element.

$$\mathbb{E}[G_{t:h}] = \mathbb{E}[p_t(R_{t+1} + \gamma G_{t+1:h})] + \mathbb{E}[(1 - p_t)V_{h-1}(S_t)] \tag{32}$$

We simplify by realizing that p_t is the only bit dealing with probability and isolated the expected value to that piece.

$$\mathbb{E}[G_{t:h}] = \mathbb{E}[p_t(R_{t+1} + \gamma G_{t+1:h})] + (1 - \mathbb{E}[p_t])V_{h-1}(S_t) \tag{33}$$

...using the law of total expectations, we can then make the following change, as $\mathbb{E}[p_t(R_{t+1} + \gamma G_{t+1:h})] = \mathbb{E}[\mathbb{E}[p_t(R_{t+1} + \gamma G_{t+1:h})|S_t, A_t]]$

$$\mathbb{E}[G_{t:h}] = \mathbb{E}[\mathbb{E}[p_t(R_{t+1} + \gamma G_{t+1:h})|S_t, A_t]] + (1 - \mathbb{E}[p_t])V_{h-1}(S_t) \tag{34}$$

We can use the definition of expected value to convert the inner expectation of the first term to the expected value of the given expression conditioned on the state-action pair...

$$\mathbb{E}[G_{t:h}] = \mathbb{E}[\frac{\pi(A_t|S_t)}{b(A_t|S_t)}(R_{t+1} + \gamma\mathbb{E}[G_{t+1:h}|S_{t+1}])] + (1 - \mathbb{E}[p_t])V_{h-1}(S_t) \tag{35}$$

$$\mathbb{E}[G_{t:h}] = \sum_{a_t}\frac{\pi(a_t|S_t)}{b(a_t|S_t)}\sum_{r_{t+1},s_{t+1}} p(r_{t+1}, s_{t+1}|S_t, a_t)[r_{t+1} + \gamma\mathbb{E}[G_{t+1:h}|S_{t+1}]] \tag{36}$$

$$\mathbb{E}[G_{t:h}] = \sum_{a_t}\pi(a_t|S_t)\sum_{r_{t+1},s_{t+1}} \frac{p(r_{t+1}, s_{t+1}|S_t, a_t)}{b(a_t|S_t)}[r_{t+1} + \gamma\mathbb{E}[G_{t+1:h}|S_{t+1}]] \tag{37}$$

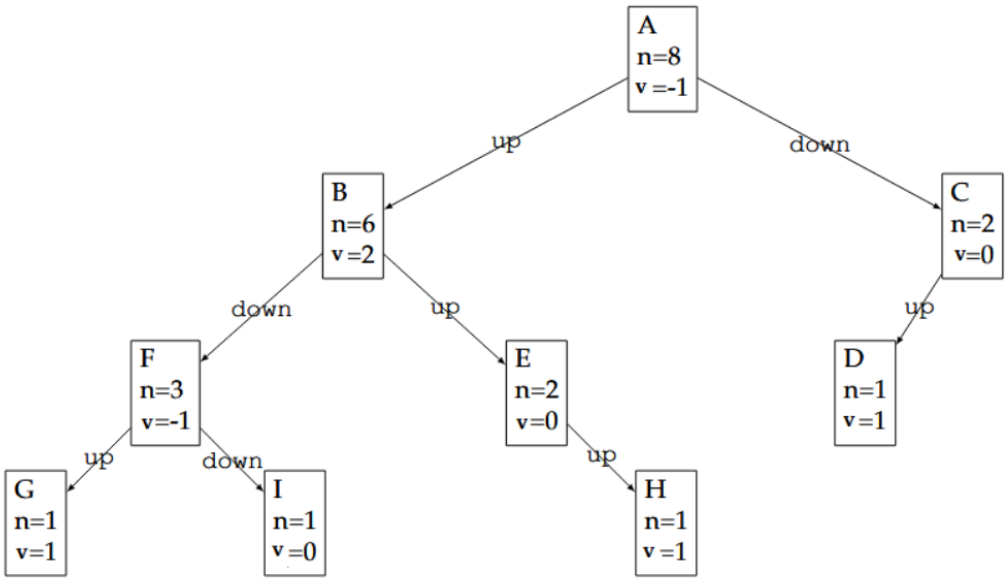

Figure 5: Problem 7: MCTS Tree

Here we see that the expected value of the return with control variates is equivalent to the expected value of the return without the control variates because our last term ends up going to zero. Because of this, we can expect p_t to go to 1:

$$\mathbb{E}[p_t] = \mathbb{E}\left[\frac{\pi(A_t|S_t)}{b(A_t|S_t)}\right] = \sum_a \pi(a|S_t)\frac{\pi(a|S_t)}{b(a|S_t)} = \sum_a \frac{\pi(a|S_t)^2}{b(a|S_t)} = \mathbb{E}\left[\frac{\pi(A_t|S_t)^2}{b(A_t|S_t)}\right] = 1 \tag{38}$$

...which shows that the introduction of a control variate did not introduce bias to the original expectation return.

Problem 6

If you were to design an algorithm called TD-search by modifying the Monte-Carlo Tree Search, how do you go about it. In particular, which part of the MCTS you would modify and how?

We would probably target the update values of nodes within the tree itself. Specificlaly, when MCTS traverses through its tree, MCTS simulates a full episode from the current state to a known terminal state. Instead we would perform a TD-backup from the current state to an estimated value of the next state, based on current value estimates and reward obtained from the next step. Value estimates of visited nodes would then be updated using this method.

Problem 7

Assume we are using Monte Carlo Tree Search (MCTS) to decide on the next action for a game with two actions at each state (up and down). The state of the tree at time t is in the accompany figure.

Each state has a name (A, B, ..), a return value v and n value. Assume that the constant c in UCB1 is 0.5.

7.1

What is the node that is selected next? Show your work.

$$UCB1(s) = \frac{v_i}{n_i} + c\sqrt{\frac{\ln N_i}{n_i}} \tag{39}$$

...wherein v_i is our simulation reward for node s, N_i is the number of visits of the parent of our node s, and n_i is the number of visits for the node s.

Let's calculate the UCB1 value for each node in our tree to determine the next state, ignoring our parent node.

- **B** - $\frac{2}{6} + 0.5\sqrt{\frac{\ln 8}{6}} = \frac{1}{3} + 0.5 * 0.589 = 0.333 + 0.294 = 0.627$
- **C** - $\frac{0}{2} + 0.5\sqrt{\frac{\ln 8}{2}} = \frac{0}{2} + 0.5 * 1.02 = 0.501$
- **F** - $\frac{-1}{3} + 0.5\sqrt{\frac{\ln 6}{3}} = \frac{-1}{3} + 0.5 * 0.773 = 0.053$
- **E** - $\frac{0}{2} + 0.5\sqrt{\frac{\ln 6}{2}} = 0.5 * 0.947 = 0.473$
- **D** - $\frac{1}{1} + 0.5\sqrt{\frac{\ln 2}{1}} = 1 + 0.5 * 0.832 = 1.416$
- **G** - $\frac{1}{1} + 0.5\sqrt{\frac{\ln 3}{1}} = 1 + 0.5 * 1.048 = 1.524$
- **I** - $\frac{0}{1} + 0.5\sqrt{\frac{\ln 3}{1}} = 0.5 * 1.048 = 0.524$
- **H** - $\frac{1}{1} + 0.5\sqrt{\frac{\ln 2}{1}} = \frac{-1}{3} + 0.5 * 0.832 = 0.083$

Since **G** has the highest calculated UCB1, we will choose that node.

7.2

What action is next for the selected node? This action will make a new state call it "J".

There is no specified rule presented - it seems to suggest, by naming, that it chooses the up action first, then later checks the down action. This could also occur in a random choice, which would also work if both actions had equivalent probability in our scenario. For now, we'll assume it's up first based on precedence and move from there.

7.3

What is the "n" and "v" value for the new state "J"?

The new node, unvisited and freshly created, is initialized with $n = 0$ and $v = 0$.

7.4

If the simulation rollout from the expanded node gives a value of 1, backup that value to all of the affected nodes

Working backwards:

- **J** - $n = 1$, $v = 1$
- **G** - $n = 2$, $v = 2$
- **F** - $n = 4$, $v = 0$
- **B** - $n = 7$, $v = 3$
- **A** - $n = 9$, $v = 0$

7.5

Assume that after this final rollout, we've run out of time to run the MCTS simulation and must now choose an action from state A. What action will be chosen and why if we use the greedy tree policy.

We would choose the "up" action moving into state **B** over **C** as it's current value at this state (3) is higher than 0.

Problem 8

In this question we are comparing mean and variance of the estimate reward for on-policy and off-policy algorithms. Consider the on-policy scenario below where actions are taken under target policy π:

Action	Reward	Probability via π
$\rightarrow$	+10	0.9
$\leftarrow$	+20	0.1

8.1

What is the expected value (mean) of estimated reward?

The expected value of our estimated reward is the sum of the total rewards multiplied by the probability of each action. Thus:

$$\sum \pi(a) * r_a \tag{40}$$

$$(0.9 * 10) + (0.1 * 20) = 11 \tag{41}$$

8.2

What is the variance of the estimate reward? Hint: use the equation $Var[X] = E[X^2] - E[X]^2$

$$E[X^2] = (0.9 * 10^2) + (0.1 * 20^2) = (0.9 * 100) + (0.1 * 400) = 130 \tag{42}$$

$$Var[X] = 130 - 11^2 = 130 - 121 = 9 \tag{43}$$

8.3

Now assume that there is a behavior policy that is taking actions with following distribution:

Action	Reward	π_t	π_b
$\rightarrow$	+10	0.9	0.5
$\leftarrow$	+20	0.1	0.5

Assume that we don't know the distribution of the target policy and we only know the ratio $\frac{\pi}{b}$ (importance sampling ratio). Numerically calculate the expected reward assuming that the action are taken by the behavior policy.

$$(0.5 * 10) + (0.5 * 20) = 15 \tag{44}$$

8.4

Calculate the variance of the expected reward?

Using $Var[X] = E[X^2] - E[X]^2$ again:

$$E[X^2] = (0.4 * 10^2) + (0.1 * 20^2) = (0.5 * 100) + (0.5 * 400) = 250 \tag{45}$$

$$Var[X] = 250 - 15^2 = 250 - 225 = 25 \tag{46}$$

8.5

What is the intuition behind higher variance in off-policy in this example? Can you explain how the importance sampling could increase the variance in off-policy learning?

In off-policy learning, the behavior policy is used to generate samples for its exploration and the target policy is derived from those samples. The behavior policy may as a result take actions that the target policy would not, leading to our observed difference in the probability distributions of actions for the target and behavior policies. This difference in distribution can cause the importance sampling ratio to be high, leading to high variance in the estimates. On-policy learning (like our first example) has a lower variance as it is more likely to result in the same actions being chosen repeatedly - which would explain the higher probability of choosing a singular action.