Q4. [14 pts] RL: Rest and ReLaxation

Consider the grid world MDP below, with unknown transition and reward functions.

A	B	C
D	E	F
G	H	I

The agent observes the following samples in this grid world:

s	a	s'	$R(s,a,s')$
E	East	F	-1
E	East	H	-1
E	South	H	-1
E	South	H	-1
E	South	D	-1

Reminder: In grid world, each non-exit action succeeds with some probability. If an action (e.g. North) fails, the agent moves in one of the cardinally adjacent directions (e.g. East or West) with equal probability, but will not move in the opposite direction (e.g. South).

Let p denote the probability that an action succeeds.

In this question, we will consider 3 strategies for estimating the transition function in this MDP.

Strategy 1: The agent does not know the rules of grid world, and runs model-based learning to directly estimate the transition function.

(a) [1 pt] From the samples provided, what is $\hat{T}$(E, South, H)?

- ○ 0
- ○ 1/5
- ○ 2/5
- ○ 3/5
- ○ 4/5
- ○ 1/3
- ● 2/3
- ○ 1/2
- ○ 1
- ○ Not enough information

There are 3 samples that move South from E, and 2 of them result in successor state H.

(b) [1 pt] From the samples provided, what is $\hat{T}$(E, West, D)?

- ○ 0
- ○ 1/5
- ○ 2/5
- ○ 3/5
- ○ 4/5
- ○ 1/3
- ○ 2/3
- ○ 1/2
- ○ 1
- ● Not enough information

We have no samples that move West from E, so there is not enough information to empirically estimate this transition probability.

Strategy 2: The agent knows the rules of grid world, and runs model-based learning to estimate p. Then, the agent uses the estimated $\hat{p}$ to estimate the transition function.

(c) [1 pt] From the samples provided, what is $\hat{p}$, the estimated probability of an action succeeding?

- ○ 0
- ○ 1/5
- ○ 2/5
- ● 3/5
- ○ 4/5
- ○ 1/3
- ○ 2/3
- ○ 1/2
- ○ 1
- ○ Not enough information

There are 3 samples of successful actions: 1 sample of E East F, and 2 samples of E South H.

The other 2 samples are unsuccessful actions: E East H, and E South D.

(d) [1 pt] Based on $\hat{p}$, what is $\hat{T}$(E, West, D)?

○ 0	● 3/5	○ 2/3	○ Not enough information
○ 1/5	○ 4/5	○ 1/2	
○ 2/5	○ 1/3	○ 1	

Even though we have never seen a sample that moves West from E, we can still use the estimated parameters in the grid world to provide an estimate. Our estimate is that actions succeed with probability 3/5, so moving West from E will succeed (and land in D) with probability 3/5.

(e) [3 pts] Select all true statements about comparing Strategy 1 and Strategy 2.

- ☐ Strategy 1 will usually require fewer samples to estimate the transition function to the same accuracy threshold.
- ☐ There are fewer unknown parameters to learn in Strategy 1.
- ■ Strategy 1 is more prone to overfitting on samples.
- ○ None of the above

Option 1: False. Estimating the transition function directly would require collecting samples for every state-action pair. By contrast, estimating the grid world parameters can be done even if you don't see every state-action pair.

Option 2: False. The transition function has probabilities for every (s, a, s') transition. By contrast, there is only one grid world parameter to estimate, the probability of an action succeeding.

Option 3: True. The transition function for some (s, a, s') transition is only estimated using the samples that start in s and take action a. If the samples for that particular state/action pair are biased, then the transition function for that value will also be biased.

The grid world and samples, repeated for your convenience:

A	B	C
D	E	F
G	H	I

s	a	s'	$R(s,a,s')$
E	East	F	−1
E	East	H	−1
E	South	H	−1
E	South	H	−1
E	South	D	−1

Strategy 3: The agent knows the rules of grid world, and uses an exponential moving average to estimate p. Then, the agent uses the estimated $\hat{p}$ to estimate the transition function.

(f) [2 pts] Consider this update equation: $\hat{p} \leftarrow (1-\alpha)\hat{p} + (\alpha)x$

Given a sample (s, a, s'), what value of x should be used in the corresponding update?

- ○ $R(s,a,s')$
- ● 1.0 if the action succeeded, and 0.0 otherwise
- ○ 1.0 if the action failed, and 0.0 otherwise
- ○ $V(s)$
- ○ $V(s')$

We're trying to estimate p, the probability of an action succeeding.

Consider a sample where the action succeeds. The estimated probability of success from that one sample is 1.0. Similarly, the estimated probability of success from a sample where the action fails is 0.0.

Note that the reward and value are not needed here, because we are not trying to estimate the action of states; instead, we are trying to estimate the probability of success.

(g) [3 pts] Select all true statements about comparing Strategy 2 and Strategy 3.

- ■ Strategy 2 gives a more accurate estimate, because it is the maximum likelihood estimate.
- □ Strategy 3 gives a more accurate estimate, because it gives more weight to more recent samples.
- ■ Strategy 3 can be run with samples streaming in one at a time.
- ○ None of the above

Option 1: True. The maximum likelihood estimate comes from the count estimate.

Option 2: False. In TD learning, we want to give weight to more recent samples because they use more accurate values in their calculation. However, when we're just estimating a probability from independent samples (whose values don't depend on the estimated values of other states), then there's no reason to give more weight to recent samples.

Option 3: True. To compute a count estimate, we need to be able to count up all the samples. The exponential moving average can be computed with each sample streaming in one at a time.

The rest of the question is independent from the previous subparts.

Suppose the agent runs Q-learning in this grid world, with learning rate $0 < \alpha < 1$, and discount factor $\gamma = 1$.

(h) [1 pt] After iterating through the samples once, how many learned Q-values will be nonzero?

○ 0 ○ 1 ● 2 ○ 3 ○ 4 ○ > 4

Q(E, East) and Q(E, South) will be nonzero.

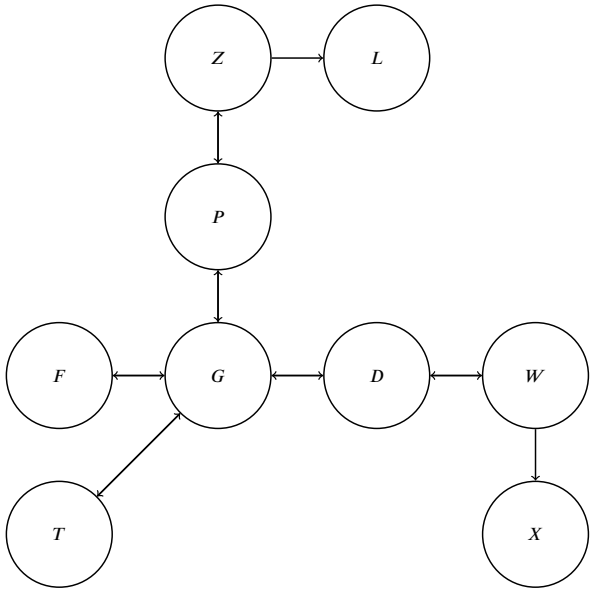
(i) [1 pt] After iterating through the samples repeatedly until convergence, how many learned Q-values will be nonzero?

○ 0 ○ 1 ● 2 ○ 3 ○ 4 ○ > 4

Still the same two nonzero values. In order to update a Q-state, we have to see a sample with that Q-state, and we only ever see two Q-states.

Q6. [12 pts] Reinforcement Learning: Island Hopping

Bob wants to traverse different islands to reach the island X. He formulates the problem as an MDP where the islands (nodes) represent the states and the arrows represent the possible actions he can take across the seas. Use the direction of the arrows (up, down, left, right) to refer to the specific actions that can be taken.



(a) [2 pts] Bob's ship doesn't always move in the direction that he wants it to. Despite this, he wants to use MLE to build an estimate of the transition function $\hat{T}$ and the reward function $\hat{R}$ for model-based reinforcement learning. He follows some specified policy and collects some data in the form of (current state, action, next state, reward) tuples shown below:

State (s)	Action (a)	New State (s')	Reward
F	Right	G	20
D	Right	G	-10
G	Up	T	-30
P	Down	Z	-15
G	Right	D	30
W	Down	D	-25
G	Right	D	30
D	Left	G	-5
G	Right	T	-30
W	Down	X	100

(i) [1 pt] What is $\hat{T}$(G, Right, D)?

$\frac{2}{3}$

Out of the 3 actions we go right from G, we end up in our desired state of D twice.

(ii) [1 pt] What is $\hat{R}$(W, Down, D)?

-25

(b) [3 pts] Bob looks to use temporal difference learning to learn the values of π, where he has the following initial values:

s	F	G	T	P	Z	L	D	W	X
$V^\pi(s)$	5	10	-4	8	2	-10	10	30	50

He performs one update step using the sample (G, Right, D, 30). Assume a discount factor $\gamma = 0.5$ and the learning rate $\alpha = 0.2$. What is the updated value of $V^\pi(G)$? Show all your work leading to your answer.

$V^\pi(G) = 15$

sample $= R(G, Right, D) + 0.5 * V^\pi(D) = 30 + 0.5 * 10 = 35$

$V^\pi(G) \leftarrow (1 - \alpha) \cdot V^\pi(G) + \alpha \cdot$ sample $= 0.8 * 10 + 0.2 * 35 = 15$

(c) [2 pts] Bob wants to have a greedy policy that will minimize exploration and maximize exploitation. Which of the following functions f will do so? Assume k is a positive real number, $N(s, a)$ represents the number of times that the state-action pair is taken, and ϵ is a very small number greater than 0.

- ☐ $f(s,a) = Q(s,a) + \frac{k}{N(s,a)}$
- ■ $f(s,a) = Q(s,a) + k \cdot e^{N(s,a)}$
- ☐ $f(s,a) = k \cdot Q(s,a) - log(N(s,a))$
- ☐ $f(s,a) = \frac{\epsilon}{k \cdot Q(s,a) \cdot N(s,a)}$
- ■ $f(s,a) = \frac{N(s,a) \cdot Q(s,a)}{\epsilon}$
- ○ None of the above

(1) - As the number of state-action pairs increases, the benefit of exploring one particular route converges to zero.

(2) - This value blows up the more times a particular route is explored, since the exponent here is positive.

(3) - Same reasoning as (1), the logarithm term will dominate the first product term as the number of state-action pairs increases.

(4) - The denominator is amplified as the number of times a particular route is explored, so the overall term will converge to zero.

(5) - Same reasoning as (4), but since the product is in the numerator and the denominator is a really small value ϵ, the function will be unbounded and converge to infinity, so one particular route will be preferred.

(d) [5 pts] Bob now switches to Q-learning, where he wants to perform approximate Q-learning for $Q(G, right)$. Assume he has w_i, which denotes the ith value of a weight vector w and $f_i(s, a)$, which denotes the value of the ith feature of the Q-state (s, a). He has the following values and observations:

State (s)	Action (a)	New State (s')	Reward (r)
G	Right	D	10
P	Up	Z	1

w_1	w_2	w_3
2	5	10

$f_1(G, Right)$	$f_2(G, Right)$	$f_3(G, Right)$
6	3	4

State	P	Z	D
Q(State, Up)	2	0	0
Q(State, Down)	5	7	0
Q(State, Left)	0	0	2
Q(State, Right)	0	-8	22

(i) [3 pts] What is the initial value of $Q(G, Right)$ based on the above weights and features?

For approximate Q-learning, we compute our desired Q-value by taking a linear combination of the weights and the corresponding features. So we have:

$Q(G, Right) = w_1 \cdot f_1(G, Right) + w_2 \cdot f_2(G, Right) + w_3 \cdot f_3(G, Right)$

$= 2 \cdot 6 + 5 \cdot 3 + 10 \cdot 4$

$= 67$

(ii) [2 pts] What is the resulting weight vector after performing the first iteration of the weight update rule for going right on G? This time, assume a discount factor $\gamma = 0.5$ and learning rate $\alpha = 0.5$.

Recall that the weight update formula for the ith weight is as follows:

$w_i \leftarrow w_i + \alpha \cdot \text{difference} \cdot f_i(s, a)$

To perform the weight update, we need to calculate the difference term. The difference term is given as:
$\text{difference} = [R(s, a, s') + \gamma \cdot max_{a'} Q(s', a')] - Q(s, a)$
where $Q(s, a)$ is the Q-value we computed using the linear combination of weights and features and $max_{a'} Q(s', a')$ denotes the maximum Q-value for the new state s' by taking a specific action a' on s'.

We compute the difference term using our observation of going right from G to D and our initial $Q(G, Right)$ from the previous part. Plugging in relevant values, we get:
$\text{difference} = [10 + 0.5 \cdot 22] - 67 = -46$

Applying this to our weight and feature vectors, we get:
$w_1 = 2 + 0.5 \cdot -46 \cdot 6 = -136$
$w_2 = 5 + 0.5 \cdot -46 \cdot 3 = 5 - 69 = -64$
$w_3 = 10 + 0.5 \cdot -46 \cdot 4 = 10 - 92 = -82$

Our final answer is $[w_1, w_2, w_3] = [-136, -64, -82]$