Information

Flag question

Feel free to use these formulae.

1. Bellam equations

$$Q(s,a) = \sum_{s'} P(s'|s,a)\left(R(s,a,s') + \gamma \max_{a'} Q(s',a')\right)$$

$$V(s) = \max_a \sum_{s'} P(s'|s,a,)\left(R(s,a,s') + \gamma V(s')\right)$$

2. Chain rule

$$P(x_1,\cdots,x_2) = P(x_n|x_{n-1},\cdots,x_1)P(x_{n-1}|x_{n-2},\cdots,x1)\cdots P(x_2|x_1)P(x_1)$$

Question **1**

Incorrect

Mark 0.00 out of 5.00

Flag question

For any MDP, changing the discount factor does not affect the optimal policy for the MDP. The this statement **True** or **False**?

Select one:

- True ✖
- False

Consider an infinite horizon setting where we have 2 states A, B, where we can alternate between A and B forever, gaining a reward of 1 each transition, or exit from B with a reward of 100. In the case that $\gamma = 1$, the optimal policy is to forever oscillate between A and B. If $\gamma = \frac{1}{2}$, then it is optimal to exit.

The correct answer is 'False'.

Question **2**

Correct

Mark 6.00 out of 6.00

Flag question

In this MDP, available actions are left, right, and stay. Stay always results in the agent staying in its current square. Left and right are successful in moving in the intended direction half of the time. The other half of the time, the agent stays in its current square. An agent cannot try to move left at the leftmost square, and cannot try to move right on the rightmost square. Staying still on a square gives a reward equivalent to the number on that square and all other transitions give zero reward (meaning any transitions in which the agent moves to a different square give zero reward).

4	0	0	36

- The value of $V_0(s)$ is 0 for all s. Perform one step of value iteration with $\gamma = 1/2$ and write $V_1(s)$ in each corresponding square:

4 ✔	0 ✔	0 ✔	36 ✔

- Perform another step of value iteration with $\gamma = 1/2$, and write $V_2(s)$ in each corresponding square:

6 ✔	1 ✔	9 ✔	54 ✔

- Using $\gamma = \frac{1}{2}$ again, write $V^*(s)$ in each corresponding square (Hint: think about states where the optimal action is obvious and work from there):

8 ✔	8 ✔	24 ✔	72 ✔

Your answer is correct.

For **Part A**:

$V_1(s)$ is equal to the optimal one-step rewards from each state. This can be achieved from each state by staying, which gives reward equal to the number on each square. (In the middle squares, moving in either direction can do no better than 0 in one step.)

For **Part B**:

From the two squares on the side, the agent can stay twice, giving $36 + \gamma \times 36 = 36 + 18 = 54$ and $4 + \gamma \times 4 = 4 + 2 = 6$. From the middle squares, the optimal policy for a two-step horizon is to move outward toward the closer side. From the middle-right, it will move right, which in two steps will give 0 reward if the first step is unsuccessful and $0 + \gamma \times 36 = 18$ if the first step is successful, giving an expectation of $\frac{1}{2} \times 0 + \frac{1}{2} \times 18 = 9$. By an identical argument, V_2 on the middle-left state is $\frac{1}{4} \times 4 = 1$.

Question **3**

Correct

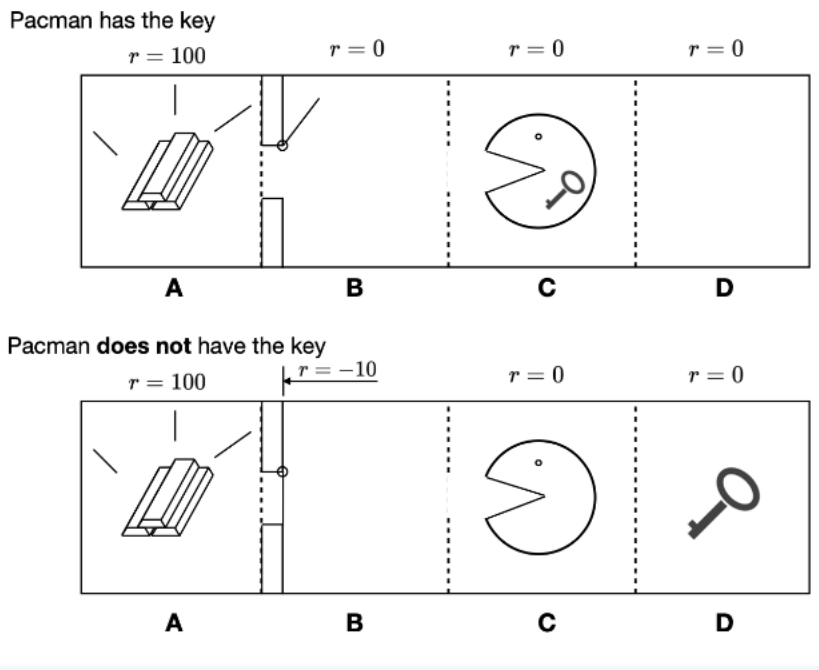
Mark 12.00 out of 12.00

Flag question

Pacman is hunting for gold in a linear grid-world with cells A, B, C, D. Cell A contains the gold but the entrance to A is locked. Pacman can pass through if he has a key. The possible states are as follows: X_k means that Pacman is in cell X and has the key; X_{-k} means that Pacman is in cell X and does not have the key. The initial state is always C_{-k}.

In each state Pacman has two possible actions, left and right. These actions are deterministic but do not change the state if Pacman tries to enter cell A without the key or run into a wall (left from cell A or right from cell D). The key is in cell D and entering cell D causes the key to be picked up instantly.

If Pacman tries to enter cell A without the key, he receives a reward of -10, i.e. $R(B_{-k}, left, B_{-k}) = -10$. The "exit" action from cell A receives a reward of 100. All other actions have 0 reward.



Consider the discount factor $\gamma = 0.1$ and the following policy:

State	A_k	B_k	C_k	D_k	A_{-k}	B_{-k}	C_{-k}	D_{-k}
Action	exit	left	left	left	exit	left	right	right

Fill in the V^{π} values for this policy below:

State	A_k	B_k	C_k	D_k	B_{-k}	C_{-k}
V	100 ✔	10 ✔	1 ✔	0.1 ✔	-11.1 ✔	0.01 ✔

Your answer is correct.

Use the formula:

$V_{\pi}(S) = R + \gamma V_{\pi}(S')\pi(a|S')$, where $\pi(a|S')$ is the deterministic policy.

Question **4**

Correct

Mark 5.00 out of 5.00

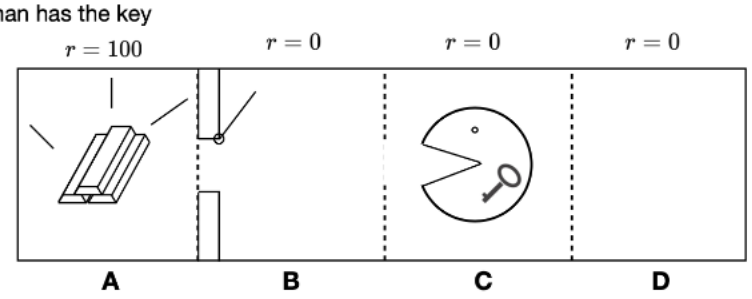
Flag question

Pacman is hunting for gold in a linear grid-world with cells A, B, C, D. Cell A contains the gold but the entrance to A is locked. Pacman can pass through if he has a key. The possible states are as follows: X_k means that Pacman is in cell X and has the key; X_{-k} means that Pacman is in cell X and does not have the key. The initial state is always C_{-k}.

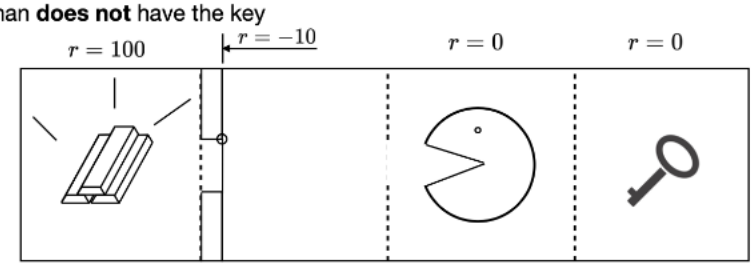
In each state Pacman has two possible actions, left and right. These actions are deterministic but do not change the state if Pacman tries to enter cell A without the key or run into a wall (left from cell A or right from cell D). The key is in cell D and entering cell D causes the key to be picked up instantly.

If Pacman tries to enter cell A without the key, he receives a reward of -10, i.e. $R(B_{-k}, left, B_{-k}) = -10$. The "exit" action from cell A receives a reward of 100. All other actions have 0 reward.

Pacman has the key



Pacman **does not** have the key



Suppose we choose to use Q-learning (in absence of the transition function) and we obtain the following observations:

s_t	a	s_{t+1}	reward
C_k	left	B_k	0
B_k	left	A_k	0
A_k	exit	terminal	100
B_{-k}	left	B_{-k}	-10

What values does the Q-function attain if we initialize the Q-values to 0 and replay the experience in the table **exactly two times**? Use a learning rate, $\alpha = 0.5$ and a discount factor, $\gamma = 0.1$

1. $Q(A_k, exit)$: ☐100 ☐50 ☐0 ☑75 ✔
2. $Q(B_k, left)$: ☑2.5 ✔ ☐0 ☐5 ☐10
3. $Q(C_k, left)$: ☑0 ✔ ☐5 ☐10 ☐2.5
4. $Q(B_{-k}, left)$: ☑-7.5 ✔ ☐5 ☐-2.5 ☐-10

Your answer is correct.

Use the formula:

$Q(s,a) \leftarrow Q(s,a) + \alpha(R(s,a) + \gamma \max_{a'} Q(s', a') - Q(s,a))$

Question **5**

Correct

Mark 5.00 out of 5.00

Flag question

Under which conditions would one benefit from using approximate Q-learning over vanilla Q- learning?

Select one:

- When the state space is very high-dimensional ✔
- When the transition function is unknown
- When the transition function is known
- When the distant factor is small

Your answer is correct.

The correct answer is: When the state space is very high-dimensional

Question 6

Partially correct

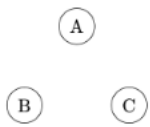
Mark 1.67 out of 5.00

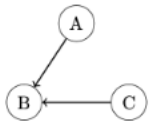
Flag question

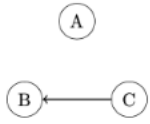
Given the joint probability table below, choose all corresponding Bayes Nets (BNs) options that can correctly represent the distribution on the right. If no such Bayes Nets are given, clearly select *None of the above*.

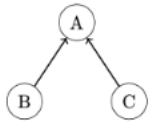
A	B	C	P(A,B,C)
0	0	0	.22
0	0	1	.08
0	1	0	.22
0	1	1	.08
1	0	0	.09
1	0	1	.11
1	1	0	.09
1	1	1	.11

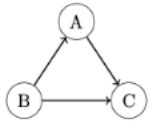
Select one or more:

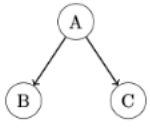












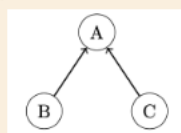
None of the above

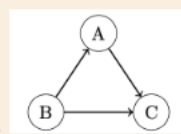
Your answer is partially correct.

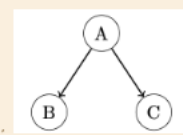
You have selected too many options.

From the table we can see that the values of $P(A,B,C)$ repeat in two blocks. This means that the value of B does not matter to the distribution. The values are otherwise all unique, meaning that there is a relationship between A and C. Together, this means that $B \perp A$, $(B \perp A|C)$, $B \perp C$, and $(B \perp C|A)$ are the only independence relationships in the distribution.

The correct answers are:







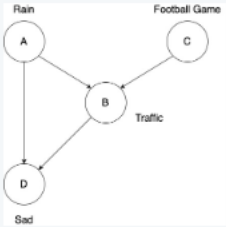
Question 7

Correct

Mark 10.00 out of 10.00

Flag question

Consider the Bayes net graph depicted below.



- Select all conditional independences that are enforced by this Bayes net graph.

☑ D⊥C | A, B

✔

☐ A⊥B

☐ D⊥C | B

☑ A⊥C

✔

☐ D⊥C

☐ A⊥C | B

☐ A⊥C | D

☐ A⊥C | B, D

- For the rest of this question, we use the original, unmodified Bayes net depicted at the beginning of the problem statement. Here are some partially-filled conditional probability tables on A, B, C, and D. Note that these are not necessarily factors of the Bayes net. Fill in the six blank entries such that this distribution can be represented by the Bayes net.

A	C	$P(C \mid A)$
$+a$	$+c$	0.8
$+a$	$-c$	0.2
$-a$	$+c$	0.8
$-a$	$-c$	0.2

A	B	D	$P(D \mid A, B)$
$+a$	$+b$	$+d$	0.60
$+a$	$+b$	$-d$	0.40
$+a$	$-b$	$+d$	0.10
$+a$	$-b$	$-d$	0.90
$-a$	$+b$	$+d$	0.20
$-a$	$+b$	$-d$	0.80
$-a$	$-b$	$+d$	0.50
$-a$	$-b$	$-d$	0.50

A	B	C	$P(C \mid A, B)$
$+a$	$+b$	$+c$	0.50
$+a$	$+b$	$-c$	0.50
$+a$	$-b$	$+c$	0.20
$+a$	$-b$	$-c$	0.80
$-a$	$+b$	$+c$	0.90
$-a$	$+b$	$-c$	0.10
$-a$	$-b$	$+c$	0.40
$-a$	$-b$	$-c$	0.60

C	$P(C)$
$+c$	(i)
$-c$	(ii)

A	B	C	D	$P(D, C \mid A, B)$
$+a$	$+b$	$+c$	$+d$	(iii)
$+a$	$+b$	$-c$	$-d$	(iv)
$+a$	$-b$	$+c$	$+d$	(v)
$+a$	$-b$	$-c$	$-d$	(vi)
⋮	⋮	⋮	⋮	⋮

i)

0.8

✔

ii)

0.2

✔

iii)

0.3

✔

iv)

0.2

✔

v)

0.02

✔

vi)

0.72

✔

Your answer is correct.

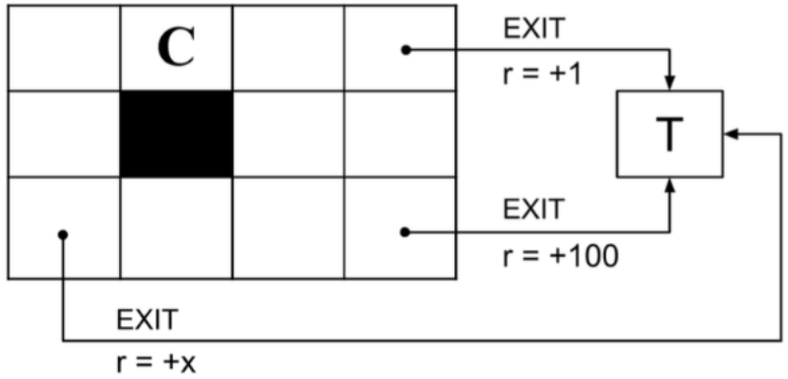
Question **8**

Correct

Mark 5.00 out of 5.00

Flag question

The following problem will take place in various instances of a grid world MDP. Shaded cells represent walls. In all states, the agent has available actions ↑, ↓, ←, →. Performing an action that would transition to an invalid state (outside the grid or into a wall) results in the agent remaining in its original state. In states with an arrow coming out, the agent has an additional action *EXIT*. In the event that the *EXIT* action is taken, the agent receives the labeled reward and ends the game in the terminal state T. Unless otherwise stated, all other transitions receive no reward, and all transitions are deterministic.



Suppose that we wanted to re-design the reward function. For which of the following new reward functions would the optimal policy **remain unchanged**? Let $R(s,a,s')$ be the original reward function.

Select one or more:

- [x] $R_2(s,a,s') = 1 + R(s,a,s')$
- [ ] $R_4(s,a,s') = -1$
- [ ] None
- [x] $R_3(s,a,s') = R(s,a,s')^2$
- [x] $R_1(s,a,s') = 10R(s,a,s')$

Your answer is correct.

R_1: Scaling the reward function does not affect the optimal policy, as it scales all Q-values by 10, which retains ordering.

R_2: Since reward is discounted, the agent would get more reward exiting then infinitely cycling between states.

R_3: The only positive reward remains to be from exiting state +100 and +1, so the optimal policy doesn't change.

R_4: With negative reward at every step, the agent would want to exit as soon as possible, which means the agent would not always exit at the bottom-right square.

The correct answers are: $R_1(s,a,s') = 10R(s,a,s')$, $R_2(s,a,s') = 1 + R(s,a,s')$, $R_3(s,a,s') = R(s,a,s')^2$

Question **9**

Partially correct

Mark 2.33 out of 5.00

Flag question

Mark **all** of the statements that are true for value iteration for any MDP.

Select one or more:

- [ ] Each iteration of value iteration produces a value function that has higher value than the prior value functions **for all states.**
- [x] Each iteration of value iteration produces a value function that has value at least as good as the prior value functions **for all states**. ✖
- [ ] Value iteration can produce a value function that has lower value than the earlier value functions **for some state**.
- [x] At convergence, the value iteration does not change the value function **for any state**. ✔
- [x] For all states s and all policies π, $V^*(s) >= V^\pi(s)$. ✔
- [ ] None of the above

Your answer is partially correct.

You have correctly selected 2.

The correct answers are:

Value iteration can produce a value function that has lower value than the earlier value functions **for some state**.

,

At convergence, the value iteration does not change the value function **for any state**.

, For all states s and all policies π, $V^*(s) >= V^\pi(s)$.

Question **10**

Partially correct

Mark 1.25 out of 5.00

Flag question

Let's assume that we don't know the transition function $T(s, a, s')$. Which of the following can be used to obtain a policy if we don't know the transition function?

Select one or more:

- [x] TD Learning (Passive) ✗
- [x] Approximate Q-learning ✓
- [ ] Policy Iteration with a learned $T(s, a, s')$
- [ ] Value Iteration followed by Policy Extraction

Your answer is partially correct.

You have correctly selected 1.

The correct answers are:

Approximate Q-learning

,

Policy Iteration with a learned $T(s, a, s')$

Question **11**

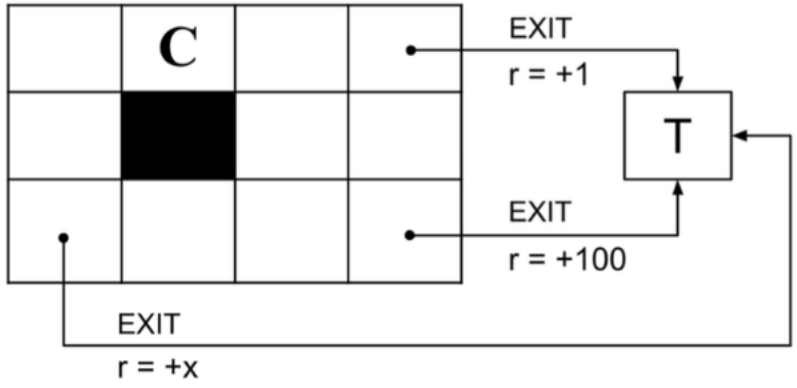
Correct

Mark 5.00 out of 5.00

Flag question

The following problem will take place in various instances of a grid world MDP. Shaded cells represent walls. In all states, the agent has available actions ↑, ↓, ←, →. Performing an action that would transition to an invalid state (outside the grid or into a wall) results in the agent remaining in its original state. In states with an arrow coming out, the agent has an additional action *EXIT*. In the event that the *EXIT* action is taken, the agent receives the labeled reward and ends the game in the terminal state T. Unless otherwise stated, all other transitions receive no reward, and all transitions are deterministic.

For all parts of the problem, assume that value iteration begins with all states initialized to zero, i.e., $V_0(s) = 0, \forall s$. Let the discount factor be $\gamma = \frac{1}{2}$.



We add a new state in which we can take the *EXIT* action with a reward of $+x$.

For what values of x is it **guaranteed** that our optimal policy π^* has $\pi^*(C) = \leftarrow$? Write the lower bound in the respective box.

Answer: x > 50 ✖

We go left if $Q(C, \leftarrow) > Q(C, \rightarrow)$. $Q(C, \leftarrow) = \frac{1}{8}x$, and $Q(C, \rightarrow) = \frac{100}{16}$. Solving for x, we get $x > 50$.

The correct answer is: 50

Comment:

Question **12**

Correct

Mark 6.00 out of 6.00

Flag question

A, B, C and D are boolean variables, and E is a random variable whose domain is e_1, e_2, e_3, e_4, e_5. Write each of the following expressions in its simplest form, i.e., a single term. Make no independence assumptions unless otherwise stated. Write "Not possible" if it is not possible to simplify the expression without making further independence assumptions.

Please keep following format rule (otherwise it will marked as wrong- we can manual grade, but if that happens because you don't pay attention, we'll deduct 1 pts per affected answer box):

- use capital P for probability

- use capitals for unfixed random variables and use lower case for fixed (observed) variables

- when there are more than one variables in either side of |, order them by alphabetical order in that side

- no space between letters and characters

- examples of accepted formats: P(A|B,C), P(a|B), P(A|B), P(A,B|C), P(a,C|B,d), P(A), P(A,B,C) etc

- example of not accepted formats:

P(A|C,B) : the B and C in the right side are not in alphabetical order

P(A | B) : there are space(s) between characters

$\Sigma_{a'} P(a' \mid B, E) P(c \mid a', B, E)$ = P(c|B,E) ✔

$\frac{\Sigma_{a'} P(B \mid a', C) P(a' \mid C) P(C)}{\Sigma_{a', e'} P(d' \mid e', C) P(e' \mid C) P(C)}$ = P(B|C) ✔

Question **13**

Partially correct

Mark 16.00 out of 18.00

Flag question

Consider the following feature based representation of the Q-function: $Q(s,a) = w_1 f_1(s,a) + w_2 f_2(s,a)$ with
$f_1(s,a) = 1/$(Manhattan distance to nearest dot after having executed action a in state s)
$f_2(s,a) =$ (Manhattan distance to nearest ghost after having executed action a in state s)

Q1:

Assume$w_1 = 1, w_2 = 10$. For the state s shown below, find the following quantities. Assume that the red and blue ghosts are both sitting on top of a dot.

Q(s, West) = 11 ✗

Q(s, South) = 11 ✓

Based on this approximate Q-function, which action would be chosen:

West ✓

Q2:

Assume Pac-Man moves West. This results in the state s' shown below. Pac-Man receives reward 9 (10 for eating a dot and -1 living penalty). Assume $\gamma = 1$.

Q(s',West)= 11 ✓

Q(s',East)= 11 ✓

sample =$[r + \gamma max_{a'} Q(s', a')]$= 20 ✓

Q3:

Now let's compute the update to the weights. Let $\alpha = 0.5$.

difference =$[r + \gamma max_{a'} Q(s', a')] - Q(s,a)$= -11 ✓

$w_1 \leftarrow w_1 + \alpha(difference) f_1(s,a)$= -4.5 ✓

$w_2 \leftarrow w_2 + \alpha(difference) f_2(s,a)$= -6.5 ✓