

CSE6423: Computer Networks I

Qualifying Exam

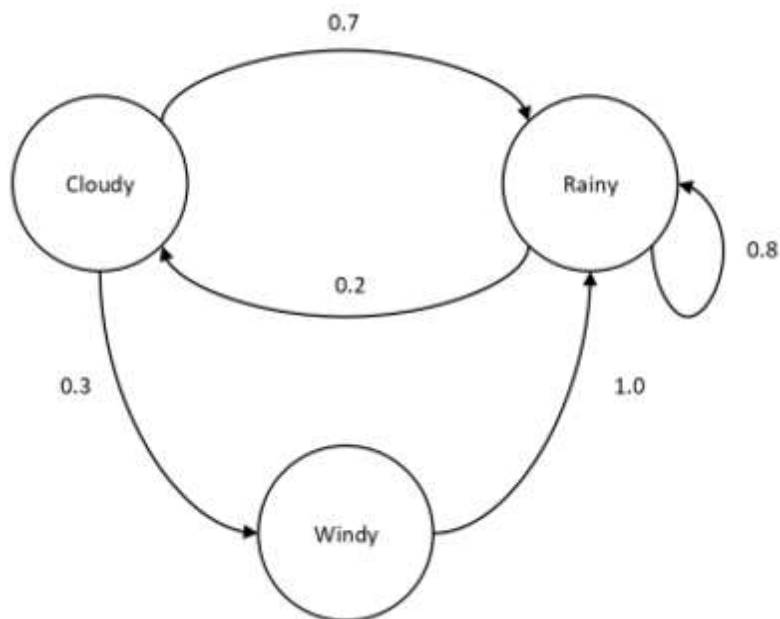
March 30, 2022

※ 모든 문제는 답만 쓰지 말고 풀이과정을 작성할 것. (답+풀이과정에 점수 부여)
풀이과정은 결과값을 계산하기 위한 과정에서 사용한 식을 함께 쓰면 됨.

1. Markov Chain

Answer the questions according to the given Markov chain.

- (1) Let us assume that today is cloudy. What is the probability that tomorrow will be rainy?
- (2) Let us assume that yesterday was cloudy, and today is rainy. What is the probability that tomorrow will be rainy again?

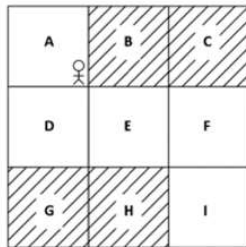


2. Value function and Q function

Answer the question below.

Value Function and Q Function

- We consider the 3x3 Grid World, where the agent starts from state A and moves to state I. (State A is the initial state, and state I is the terminal state.)
- In each state, the agent has two actions: **right** and **down**.
- The agent moves to the next state according to its action with 100% probability; the environment is not slippery.
- If the agent moves to an unshaded state, it receives a **+1** reward.
- If the agent moves to a shaded state, it receives a **-1** reward.
- If the agent moves according to the policy shown in the table, **what is the value of state A, $V(A)$** ? (discount factor is 1.)



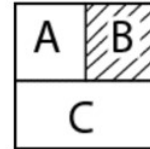
State	Action
A	50% down, 50% right
B	down
C	down
D	50% down, 50% right
E	50% down, 50% right
F	down
G	right
H	right

(States A, D, E, F, I are unshaded states. State B, C, G, H are shaded states.)

3. Value Iteration

Below is an example of using **value iteration**. Answer the question at the end.

- We have a Grid world environment with 3 states.
 - The initial state is state A, and the agent should move to state C without visiting the shaded state B.
 - We have two actions: 0 and 1.
 - The model dynamics is shown in the table below.



- Model dynamics of state A

s	a	s'	$P(s' s, a)$	$R(s, a, s')$
A	0	A	0.3	0
A	0	B	0.5	-1
A	0	C	0.2	1
A	1	A	0.3	0
A	1	B	0.1	-1
A	1	C	0.6	1

- Our goal is to calculate the value of states using **value iteration**.
- Initially, we set the value of all states to zero.

State	Value
A	0
B	0
C	0

- In the first iteration, we use the initial value table to calculate Q values for all state-action pairs.

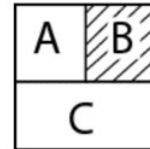
- (1) What is $Q(A, 0)$?
- (2) What is $Q(A, 1)$?
- (3) What is $V(A)$ after first iteration?

We assume that the **discount factor is 1.0**.

4. Policy Iteration

Below is an example of using **policy iteration**. Answer the question at the end.

- We have a Grid world environment with 3 states.
 - The initial state is state A, and the agent should move to state C without visiting the shaded state B.
 - We have two actions: 0 and 1.
 - The model dynamics is shown in the table below.



- Model dynamics of state A

s	a	s'	$P(s' s, a)$	$R(s, a, s')$
A	0	A	0.3	0
A	0	B	0.5	-1
A	0	C	0.2	1
A	1	A	0.3	0
A	1	B	0.1	-1
A	1	C	0.6	1

- Our goal is to calculate the value of states using **policy iteration**.
- Initially, we set the value of all states to zero. (left table)
- Also, we choose an initial policy. (right table)

State	Value
A	0
B	0
C	0

State	Action
A	1
B	0
C	1

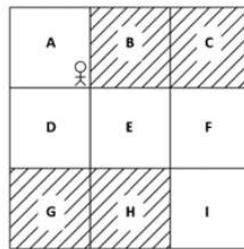
- In the first step, we use the initial policy and the initial value table to update state values iteratively.
- What is $V(A)$ after first iteration of the first step?

We assume that the **discount factor is 1.0**.

5. Monte Carlo Method

This is a problem on Monte Carlo prediction. Answer the questions at the end.

- We consider the 3x3 Grid World, where the agent starts from state A and moves to state I. (State A is the initial state, and state I is the terminal state.)
- In each state, the agent has two actions: **right** and **down**.
- If the agent moves to an unshaded state, it receives a **+1** reward.
- If the agent moves to a shaded state, it receives a **-1** reward.



- We use **Monte Carlo prediction** to predict value of the states given a policy.
- The agent has a policy, and it moves according to the **same policy in every episode**.
- However, the **environment is stochastic**, so the trajectory can be different in every episode.
- The agent ran 10 episodes, and the trajectories were the following.

Episode	T_0 (initial)	T_1	T_2	T_3	T_4 (final)
1	A	B	E	H	I
2	A	D	G	H	I
3	A	D	E	F	I
4	A	B	C	F	I
5	A	D	E	H	I
6	A	B	C	F	I
7	A	B	E	F	I
8	A	D	G	H	I
9	A	B	E	H	I
10	A	D	E	F	I

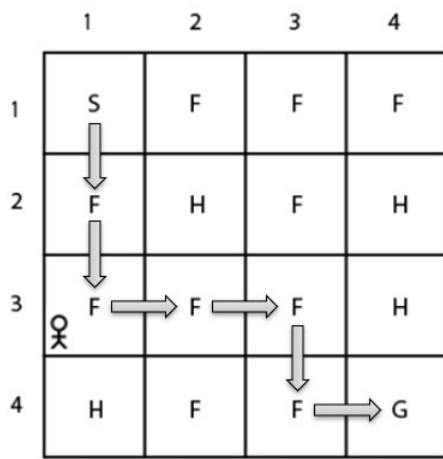
- (1) What is the value of state A, $V(A)$?
- (2) What is the value of state B, $V(B)$?
- (3) What is the value of state E, $V(E)$?
- (4) What is the value of state H, $V(H)$?

6. SARSA

This is a problem on SARSA. Answer the question at the end.

SARSA

- In SARSA, we use an epsilon-greedy policy as the behavior policy b . Suppose we run an episode according to the epsilon-greedy policy, and the trajectory of the agent is as shown below.
- When the agent is in state (3,1), the Q table is as shown in the right.



State	Action	Value
...
(3,1)	LEFT	0.2
(3,1)	DOWN	0.1
(3,1)	RIGHT	0.7
(3,1)	UP	0.3
(3,2)	LEFT	0.2
(3,2)	DOWN	0.8
(3,2)	RIGHT	0.6
(3,2)	UP	0.3
...

We assume that the environment is non-slippy (deterministic). +1 reward is given when the agent enters state (4,4). Otherwise, the reward is 0. The learning rate $\alpha=0.1$, and the discount factor $\gamma=1.0$.

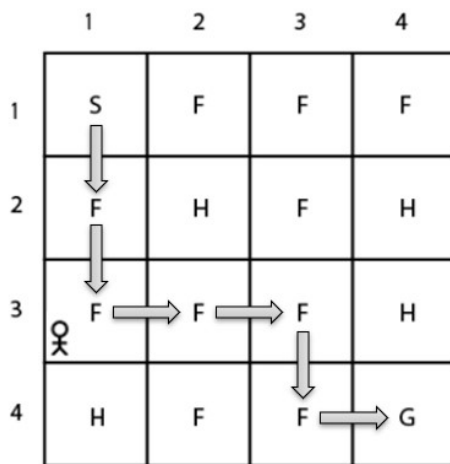
After this episode, what is the new value of $Q((3,1), \text{right})$?

7. Q-learning

This is a problem on Q-learning. Answer the question at the end.

Q-Learning

- In Q-learning, we use an epsilon-greedy policy as the behavior policy b . Suppose we run an episode according to the epsilon-greedy policy, and the trajectory of the agent is as shown below.
- When the agent is in state (3,1), the Q table is as shown in the right.



State	Action	Value
...
(3,1)	LEFT	0.2
(3,1)	DOWN	0.1
(3,1)	RIGHT	0.7
(3,1)	UP	0.3
(3,2)	LEFT	0.2
(3,2)	DOWN	0.8
(3,2)	RIGHT	0.6
(3,2)	UP	0.3
...

We assume that the environment is non-slippy (deterministic). +1 reward is given when the agent enters state (4,4). Otherwise, the reward is 0. The learning rate $\alpha=0.1$, and the discount factor $\gamma=1.0$.


After this episode, what is the new value of $Q((3,1), \text{right})$?

8. TD Prediction

This is a problem on TD prediction. Answer the questions at the end.

TD Prediction

- We consider the **non-slippy (deterministic)** Frozen Lake environment. We are going to predict the state values given a policy, using TD prediction.
- The given policy is shown in the right table. (Actions are not shown for terminal states.)
- The initial state is (1,1) and the goal state is (4,4).

	1	2	3	4
1	S 	F	F	F
2	F	H	F	H
3	F	F	F	H
4	H	F	F	G

policy			
RIGHT	RIGHT	DOWN	LEFT
DOWN	-	DOWN	-
RIGHT	DOWN	DOWN	-
-	RIGHT	RIGHT	-

- Initially, we set all state values to zero.

0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0

- From the initial table, we use the **TD learning update rule** to update the table as we run the episodes.

$$V(s) = V(s) + \alpha(r + \gamma V(s') - V(s))$$

We assume $\alpha=0.1$, and $\gamma=1.0$.

- (1) What is **$V(4,3)$** after the first episode?
- (2) What is **$V(4,3)$** after the second episode?
- (3) What is **$V(4,3)$** after the third episode?

+1 reward is given when the agent enters state (4,4). Reward is 0 for all other cases.