

# COMPSCI 762 Tutorial 9

## Tutorial on Reinforcement Learning and Association Rule Mining

---

Luke Chang

May 2021

The University of Auckland

Reinforcement Learning

Association Rule Mining

# Reinforcement Learning

---

# Terminology

- **Agent:** A hypothetical entity which performs actions in an environment to gain some reward.
- **Environment:** A scenario the agent has to face.
- **Action**  $a_t \in A$ : All the possible moves that the agent can take.
- **State**  $s_t \in S$ : Current situation returned by the environment.
- **Reward**  $R(s_t, a_t)$ : An immediate return sent back from the environment to evaluate the last action by the agent.
- **Policy**  $\pi : S \rightarrow A$ : The strategy that the agent employs to determine next action based on the current state.
- **Value**  $V^\pi(s)$ : The expected long-term return with discount  $\gamma$ , as opposed to the short term reward  $R$ .  $V^\pi(s)$  is defined as the expected long term return of the current state  $s$  under policy  $\pi$ .
- **Q-value, action-value**  $Q^\pi(s, a)$ : is similar to **Value**, except it takes the current action  $a$ .  $Q^\pi(s, a)$  refers to the long term return of the current state  $s$ , taking action  $a$  under policy  $\pi$ .

# Markov Decision Process (MDP)

MDP is defined by  $(S, A, R, \mathbb{P}, \gamma)$

- $S$ : Set of possible states  $s_t \in S$
- $A$ : Set of possible actions  $a_t \in A$
- $R$ : Immediate reward given by the state and action pair  $R(s_t, a_t)$
- $\mathbb{P}$ : Transition probability at state  $s$  if an action  $a$  is taken
- $\gamma$ : Discount factor with  $0 \leq \gamma < 1$ ; The weight of future rewards

## Markov Property:

$$P(s_{t+1} | s_t, a_t, s_{t-1}, a_{t-1}, s_{t-2}, a_{t-2}, \dots) = P(s_{t+1} | s_t, a_t)$$

A *stochastic process*, where the future is solely determined by the current state and action.  
The past and the future are independent.

# Markov Decision Process

- Assume the reward  $r_t$  the Markov property:

$$P(r_t | \langle s_t, a_t \rangle, \langle s_{t-1}, a_{t-1} \rangle, \langle s_{t-2}, a_{t-2} \rangle, \dots) = P(r_t | s_t, a_t)$$

Immediate reward  $r_t$  is solely based on the current state and action pair  $R(s_t, a_t)$ .

- The task:** Learn a policy  $\pi : S \rightarrow A$  to maximize the expected current and future rewards

$$\mathbb{E}[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \dots]$$

for every possible starting state  $s_0$ .

- Intuition:** How we got here doesn't matter, what is the current best move?

## Escape the grid-world: Solving an MDP - Value Iteration

Your task is to design the AI to help a robot to escape the room. The door is at right top corner. The actions are {Left, Up, Right, Down}.

+10 ↑		
0	0	0
0	0	0

Initially, no action is taken, we set the value to 0 for all states.

## Value Iteration - 1 Action

If we can only take 1 action in the game.

0	0	10
0	0	0

↑  
+10



## Value Iteration - 2 Actions

When we take more than one action, we have to balance immediate reward and future reward.  $\gamma$  controls the importance of future rewards.

Let  $\gamma = 0.5$ , state values with 2 actions:

$$\begin{aligned} V^\pi(s) &= \mathbb{E}[\sum_{t \geq 0} \gamma^t r_t] = r_0 + \gamma r_1 + \gamma^2 r_2 + \dots \\ &= 0 + 0.5 \times 10 = 5 \end{aligned}$$

+10 ↑		
0	5	10
0	0	5

## Value Iteration - 3 Actions

Let  $\gamma = 0.5$ , state values with 3 actions:

$$V^\pi(s) = 0 + 0 + 0.5^2 \times 10 = 2.5$$

		+10 ↑
2.5	5	10
0	2.5	5

# Value Iteration

Let  $\gamma = 0.5$ , state values with 4 actions:

$$V^\pi(s) = 0 + 0 + 0 + 0.5^3 \times 10 = 1.25$$

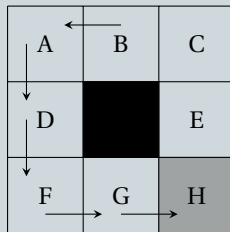
			+10
			↑
2.5	5	10	
1.25	2.5	5	

**Caveat:** Do not mix value  $V^\pi(s)$  and reward  $R(s_t, a_t)$ .

## 361 Exam Question 8, 2020

### Example

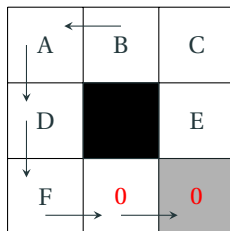
- The black cell cannot be entered.
- The actions are {Left, Up, Right, Down}.
- The reward for actions that bring it into the target is 0. Other actions get a reward of  $-1$ .
- The partial policy,  $\pi$ , is given by the arrows in the grid.
- Discount factor,  $\gamma = 0.5$



## 361 Exam Question 8, 2020

- The reward for actions that bring it into the target is 0. Other actions get a reward of  $-1$ .
- The partial policy,  $\pi$ , is given by the arrows in the grid.
- $\gamma = 0.5$

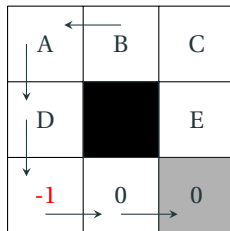
Give for each state the value of the value function,  $V$ , for the given policy. You can ignore states for which no policy is defined.



- The target state, H, requires 0 action,  $V^\pi(s = H) = 0$ . Note: There is no action called “stay”.
- State G requires 1 action,  $V^\pi(s = G) = 0$

## 361 Exam Question 8, 2020

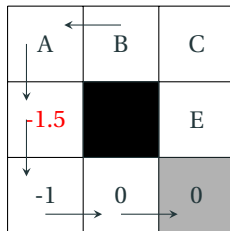
- The reward for actions that bring it into the target is 0. Other actions get a reward of  $-1$ .
- The partial policy,  $\pi$ , is given by the arrows in the grid.
- $\gamma = 0.5$



- State F requires 2 action,  $V^\pi(s = F) = -1 + 0.5 \times 0 = -1$

## 361 Exam Question 8, 2020

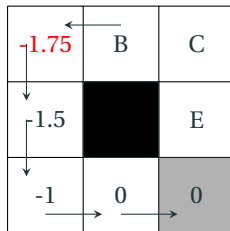
- The reward for actions that bring it into the target is 0. Other actions get a reward of  $-1$ .
- The partial policy,  $\pi$ , is given by the arrows in the grid.
- $\gamma = 0.5$



- State D requires 3 action,  $V^\pi(s = D) = -1 + 0.5 \times (-1) + 0.5^2 \times 0 = -1.5$

## 361 Exam Question 8, 2020

- The reward for actions that bring it into the target is 0. Other actions get a reward of  $-1$ .
- The partial policy,  $\pi$ , is given by the arrows in the grid.
- $\gamma = 0.5$

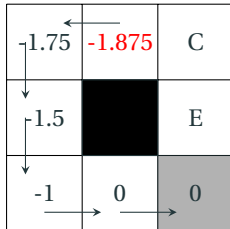


- State A requires 4 action,  $V^\pi(s = A) = -1 + 0.5 \times (-1) + 0.5^2 \times (-1) + 0.5^3 \times 0 = -1.75$



## 361 Exam Question 8, 2020

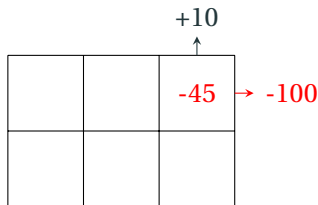
- The reward for actions that bring it into the target is 0. Other actions get a reward of  $-1$ .
- The partial policy,  $\pi$ , is given by the arrows in the grid.
- $\gamma = 0.5$



- State B requires 5 action,  
$$V^\pi(s = A) = -1 + 0.5 \times (-1) + 0.5^2 \times (-1) + 0.5^3 \times (-1) + 0.5^4 \times 0 = -1.875$$

## Limitations on MDP - The Cliff Sernario

- The value of a state is the expected reward from taking the best action in the state.
- Accumulating rewards from random actions would calculate the expected reward from random actions in the state.
- E.g.: We would learn that any state near a cliff is bad, because you get a negative score if you jump off, even though you don't have to jump off.



**Intuition:** Instead of computing value  $V^\pi(s)$ , we learn **Q-value**,  $Q^\pi(s, a)$ , which considers the state  $s$  and the action  $a$  as a pair.

- Q-learning can identify an **optimal action-selection policy** for any given a finite Markov decision process (FMDP).
- $Q : S \times A \rightarrow \mathbb{R}$ , calculating the quality of a state–action combination
- Iterative method,  $Q$  is initialized to an arbitrary fixed value. At each time  $t$ , the agent selects an action  $a_t$ , observes a reward  $r_t$ , enters a new state  $s_{t+1}$ , and  $Q$  is updated.

The Q-value is updated by:

$$Q^{\text{new}}(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \cdot [R(s_t, a_t) + \gamma \cdot \max_{a'} Q(s_{t+1}, a') - Q(s_t, a_t)]$$

where  $\alpha$  is the learning rate. (In a fully deterministic environment, a learning rate of  $\alpha = 1$  is optimal.)

## Example

- For the transitions into non-existing cell the next state,  $s'$ , is equal to the current state,  $s$ .
- For each action, the agent gets a reward of  $-1$ . The reward for actions that bring it into the target is  $0$ , and the black state is  $-2$ .
- Discount factor,  $\gamma = 0.5$
- Let the initial value of  $Q$  be  $-16$ .

A	B	C
D		E
F	G	H

## Q-Learning: 361 Exam Question 8, 2020

A	B	C
D		E
-16 -16 -16 -16 -16 →	-16 -16 -16 -16 -16	H

# Association Rule Mining

---



# Apriori Algorithm



