COMPSCI 762 Tutorial 9

Tutorial on Reinforcement Learning and Association Rule Mining

Luke Chang

May 2021

The University of Auckland

Topics

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 $\pi : S \to A$:** 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 γ, 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 π.
- **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 π.

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
- γ: Discount factor with $0 \leq \gamma < 1$; The weight of future rewards

Markov Property:

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

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:

$$P(r_t| < s_t, a_t >, < s_{t-1}, a_{t-1} >, < s_{t-2}, a_{t-2} >, \ldots) = P(r_t|s_t, a_t)$$

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

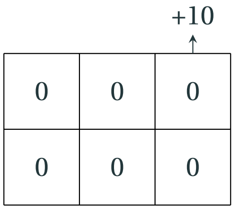
- **The task:** Learn a policy $\pi : S \rightarrow A$ to maximizes the expected current and future rewards

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

  for every possible starting state s_0.

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

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 $\{\text{Left}, \text{Up}, \text{Right}, \text{Down}\}$.



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.

+10

0	0	10
0	0	0

Value Iteration - 2 Actions

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

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

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

+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

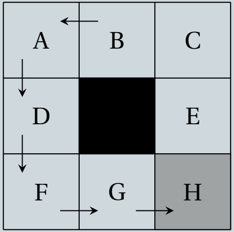
2.5	5	10
1.25	2.5	5

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

361 Exam Question 8, 2020

Example

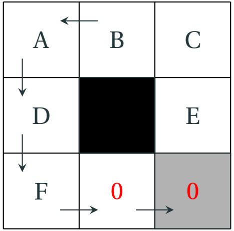
- The black cell cannot be entered.
- The actions are $\{\text{Left}, \text{Up}, \text{Right}, \text{Down}\}$.
- The reward for actions that bring it into the target is 0. Other actions get a reward of -1.
- The partial policy, π, is given by the arrows in the grid.
- Discount factor, $\gamma = 0.5$



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, π, is given by the arrows in the grid.
- $\gamma = 0.5$

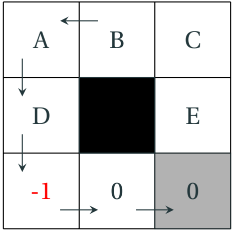
Give for each state the value of the value function, V, for the given policy. You can ignore states for which no policy is defined.



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

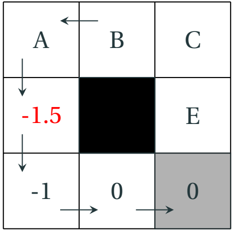
361 Exam Question 8, 2020

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



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

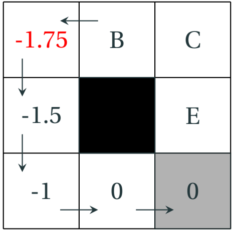
- The reward for actions that bring it into the target is 0. Other actions get a reward of -1.
- The partial policy, π, is given by the arrows in the grid.
- $\gamma = 0.5$



- 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

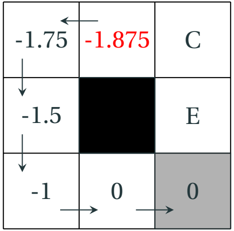
- The reward for actions that bring it into the target is 0. Other actions get a reward of -1.
- The partial policy, π, is given by the arrows in the grid.
- $\gamma = 0.5$



- 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

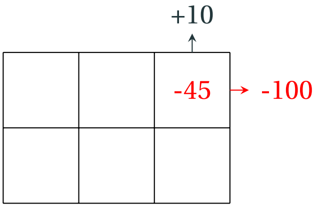
- The reward for actions that bring it into the target is 0. Other actions get a reward of -1.
- The partial policy, π, is given by the arrows in the grid.
- $\gamma = 0.5$



- 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.



Q-Learning

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).
- $Q: S \times A \to \mathbb{R}$, 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 reword r_t, enters a new state s_{t+1}, and Q is updated.

The Q-value is updated by:

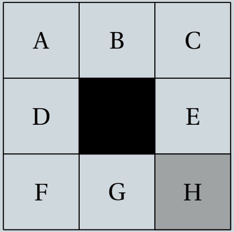
$$Q^{\text{new}}(s_t,a_t) \leftarrow Q(s_t,a_t) + \alpha \cdot [R(s_t,a_t) + \gamma \cdot \max_{a'} Q(s_{t+1},a') - Q(s_t,a_t)]$$

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

Q-Learning: 361 Exam Question 8, 2020

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.



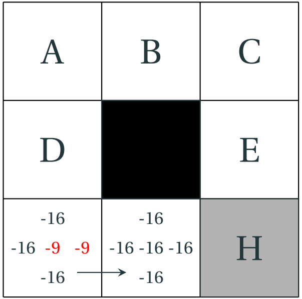
Q-Learning: 361 Exam Question 8, 2020

Let the agent walks the path (F, right) → (G, up) → (G, right). Calculate values for the Q-table after every step of the path. What is the value of $V(G)$ after the last step?

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

Q-Learning: 361 Exam Question 8, 2020

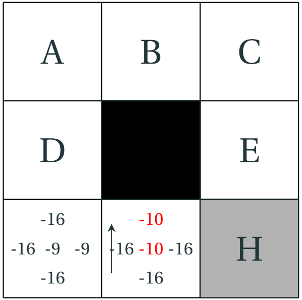
Let the agent walks the path (F, right) $\rightarrow$ (G, up) $\rightarrow$ (G, right). Calculate values for the Q-table after every step of the path. What is the value of $V(G)$ after the last step?



Given $\gamma = 0.5$, $Q(s = F, a = \text{right}) = -1 + 0.5 \times (-16) = -9$

Q-Learning: 361 Exam Question 8, 2020

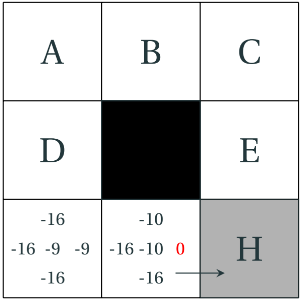
Let the agent walks the path (F, right) $\rightarrow$ (G, up) $\rightarrow$ (G, right). Calculate values for the Q-table after every step of the path. What is the value of $V(G)$ after the last step?



$$Q(G, \text{up}) = -2 + 0.5 \times (-16) = -10$$

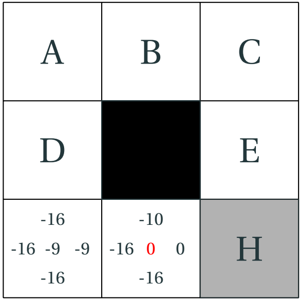
Q-Learning: 361 Exam Question 8, 2020

Let the agent walks the path (F, right) $\rightarrow$ (G, up) $\rightarrow$ (G, right). Calculate values for the Q-table after every step of the path. What is the value of $V(G)$ after the last step?



$$Q(G, \text{right}) = 0 + 0.5 \times 0 = 0$$

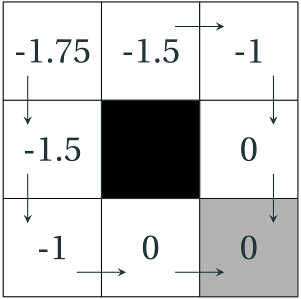
Let the agent walks the path (F, right) $\rightarrow$ (G, up) $\rightarrow$ (G, right). Calculate values for the Q-table after every step of the path. What is the value of $V(G)$ after the last step?



$$V(G) = 0$$

Q-Learning: 361 Exam Question 8, 2020

Create an optimal policy π^*, and draw it on the given diagram. What are the corresponding values of V^*? What is the most desirable state for the agent?



Note: The initial value of Q does not affect $V(s)$.

Association Rule Mining

Terminology

- **Itemset:** A collection of one or more items, e.g. $X = \{A, B\}$, $Y = \{B\}$ Note: Single item can be an itemset.
- $\boldsymbol{N = |T|}$**:** is the number of transactions (instances)
- $\boldsymbol{d = |I|}$**:** is the number of distinct (unique) items. There are 2^d possible itemsets.
- **Width** $\boldsymbol{w}$**:** The transaction width is the number of items present in a transaction.
- **Support count** $\boldsymbol{\sigma}$**:** Frequency of occurrence of an itemset, e.g. $\sigma(\{A, B\}) = 2$ means 2 transactions contain the itemset $\{A, B\}$.
- **Frequent Itemset:** An itemset whose support is greater than or equal to the *minsup* threshold
- **Support** $s(X \to Y)$**:** Fraction of transactions that contain an itemset

$$s(X \to Y) = \frac{\sigma(X \cup Y)}{|T|}$$

- **Confidence** $\boldsymbol{c \to Y}$**:** Measures how often items in Y appear in transactions that contain X

$$c(X \to Y) = \frac{\sigma(X \cup Y)}{\sigma(X)}$$

Apriori Algorithm

- **Goal:** Reducing the number of candidates
- **Apriori principle:** "If an itemset is frequent, then all of its subsets must also be frequent." – anti-monotone property of support

$$\forall X, Y : (X \subseteq Y) \Rightarrow s(X) \geq s(Y)$$

- **Maximal Frequent Itemset:** If none of its immediate supersets is frequent
- **Closed Frequent Itemset:** An itemset X is closed, if none of its immediate supersets has the same support as the itemset X.

$$\text{Maximal Frequent Itemset} \subseteq \text{Closed Frequent Itemset} \subseteq \text{Frequent Itemset}$$

Review Question 1 - Apriori Algorithms

Given the following transaction database, considering using Apriori algorithm to find all frequent itemsets a support threshold of 30%.

TID	**Items**
T1	**1, 2, 5**
T2	**2, 4**
T3	**2, 3**
T4	**1, 2, 4**
T5	**1, 3**
T6	**2, 3**
T7	**1, 3**
T8	**1, 2, 3, 5**
T9	**1, 2, 3**
T10	**1, 2, 5, 6**

Table 1: Candidate 1-itemsets

Item	Support Count σ
1	7
2	8
3	6
4	2
5	3
6	1

- 10 transactions. The frequent itemset must occur in at least 3 transactions.
- {1}, {2}, {3} and {5} are frequent 1-itemsets.

Review Question 1 - Frequent itemsets

Table 2: Candidate 2-itemsets

Item	σ
1, 2	7
1, 3	4
1, 5	3
2, 3	4
2, 5	3
3, 5	1

- $\{3,5\}$ is not a frequent 2-itemset.

FP-Tree