# 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 $\gamma$, 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:**

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

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 $\pi : S \to A$ 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 {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.

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. $\gamma$ controls the importance of future rewards.

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

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

+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:

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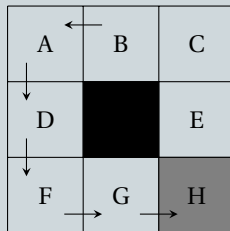
+10

| | | |
|------|-----|----|
| 2.5 | 5 | 10 |
| 1.25 | 2.5 | 5 |

**Caveat:** Do not mix value $V^{\pi}(s)$ and reward $R(s_t, a_t)$.
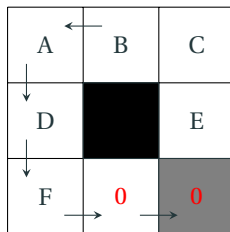
# 361 Exam Question 8, 2020

## 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, $\gamma = 0.5$

- 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, $V^{\pi}(s = G) = 0$

- 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.
- $\gamma = 0.5$
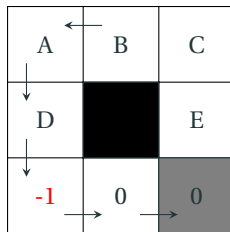


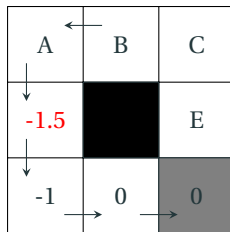- State F requires 2 action, $V^\pi(s = F) = -1 + 0.5 \times 0 = -1$

- 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, $V^{\pi}(s = D) = -1 + 0.5 \times (-1) + 0.5^2 \times 0 = -1.5$

- 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, $V^{\pi}(s = A) = -1 + 0.5 \times (-1) + 0.5^2 \times (-1) + 0.5^3 \times 0 = -1.75$

- 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.
- $\gamma = 0.5$



| | | |
|---|---|---|
| -1.75 | -1.875 | C |
| -1.5 | | E |
| -1 | 0 | 0 |

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

- 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 <u>random actions</u> 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.
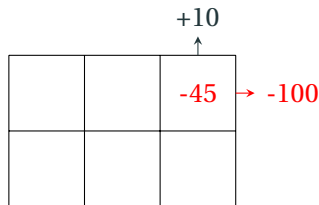
## Q-Learning

**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 $\alpha$ 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 |

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?

| A | B | C |
|---|---|---|
| D | ■ | E |
| -16<br>-16 -16 -16<br>-16 | -16<br>-16 -16 -16<br>-16 | H |

Let the agent walks the path (F, right) $\to$ (G, up) $\to$ (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 $\gamma = 0.5$, $Q(s = F, a = \text{right}) = -1 + 0.5 \times (-16) = -9$

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?



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

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<br>-16 -9 -9<br>-16 | -10<br>-16 -10 0<br>-16 → | H |

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

Let the agent walks the path (F, right) $\to$ (G, up) $\to$ (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<br>-16 -9 -9 | -10<br>-16 0 0<br>-16 | H |

$$V(G) = 0$$

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

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

Create an optimal policy $\pi^*$, and draw it on the given diagram. What are the corresponding values of $V^*$? What is the most desirable state for the agent?
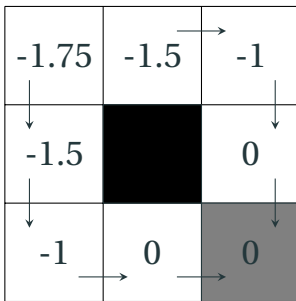
# Association Rule Mining

## Association Rule Mining

**Objective:** Given a set of transactions, find rules that predict the occurrence of an item based on the occurrences of other items in a transaction.

E.g. Given you like "Doctor Strange" and "Captain Marvel", a recommendation system predicts you will also like "Avengers:Endgame".

**Association Rule:** An implication expression of the form $X \rightarrow Y$, where $X$ and $Y$ are itemsets.

## Terminology

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

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

- **Confidence $c(X \rightarrow Y)$:** Measures how often items in $Y$ appear in transactions that contain $X$

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

# Frequent Itemset Generation – 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 $X$.

  Maximal Frequent Itemset $\subseteq$ Closed Frequent Itemset $\subseteq$ Frequent Itemset

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** |

| Item | Support Count, $\sigma$ |
|------|------------------------|
| 1 | 7 |
| 2 | 8 |
| 3 | 6 |
| 4 | 2 |
| 5 | 3 |
| 6 | 1 |

**Table 1:** Candidate 1-itemsets

- 10 transactions. The frequent itemset must occur in at least 3 transactions.

- {1}, {2}, {3} and {5} are frequent 1-itemsets.

| Item | $\sigma$ |
|------|----------|
| 1, 2 | 5 |
| 1, 3 | 4 |
| 1, 5 | 3 |
| 2, 3 | 4 |
| 2, 5 | 3 |
| 3, 5 | 1 |

**Table 2:** Candidate 2-itemsets

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

| Item | $\sigma$ |
|---------|----------|
| 1, 2, 3 | 2 |
| 1, 2, 5 | 3 |
| 1, 3, 5 | 1 |
| 2, 3, 5 | 1 |

**Table 3:** Candidate 3-itemsets

- Since $\{3, 5\}$ is not a frequent 2-itemset, any superset of $\{3, 5\}$ are NOT frequent itemset.
- Only $\{1, 2, 5\}$ is a frequent 3-itemset.
- None of 4-itemsets is frequent.

Find all association rules from frequent 3-itemset with the confidence threshold of 80%.

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

- Only $\{1,2,5\}$ is a frequent 3-itemset. Remove other 3-itemsets.
- $\sigma(X \cup Y) = \sigma(\{1,2,5\}) = 3$

| $X \rightarrow Y$ | $\sigma(X)$ | $c(X \rightarrow Y)$ |
|---|---|---|
| $\{1\} \rightarrow \{2,5\}$ | 7 | $3/7 \approx 0.43$ |
| $\{2\} \rightarrow \{1,5\}$ | 8 | $3/8 = 0.375$ |
| $\{5\} \rightarrow \{1,2\}$ | 3 | $3/3 = 1$ |
| $\{1,2\} \rightarrow \{5\}$ | 5 | $3/5 = 0.6$ |
| $\{1,5\} \rightarrow \{2\}$ | 3 | $3/3 = 1$ |
| $\{2,5\} \rightarrow \{1\}$ | 3 | $3/3 = 1$ |

$\{5\} \rightarrow \{1,2\}$, $\{1,5\} \rightarrow \{2\}$ and $\{2,5\} \rightarrow \{1\}$ are the 3-itemsets, such that $c(X \rightarrow Y) \geq minconf$.

## Review Question 2 – FP-Tree

Considering the same transactions above, use frequent-pattern (FP) growth algorithm to perform frequent itemsets mining with a support threshold of 30%.

1. What is the item head table? (Hint: keep in mind the support count and sort order.)
2. What is the FP-tree corresponding to the transactions above?

| Item | $\sigma$ | Node Link |
|------|----------|-----------|
| **2** | 8 | |
| **1** | 7 | |
| **3** | 6 | |
| **5** | 3 | |

**Table 4:** Header Table

$\{\}$

*Node Links* are omitted. They are pointers which point to the node with the corresponding TID.

## Review Question 2 – FP-Tree

Considering the same transactions above, use frequent-pattern (FP) growth algorithm to perform frequent itemsets mining with a support threshold of 30%.

1. What is the item head table? (Hint: keep in mind the support count and sort order.)
2. What is the FP-tree corresponding to the transactions above?

| Item | $\sigma$ | Node Link |
|:----:|:--------:|:----------|
| **2** | 8 | |
| **1** | 7 | |
| **3** | 6 | |
| **5** | 3 | |

**Table 4:** Header Table



*Node Links* are omitted. They are pointers which point to the node with the corresponding TID.
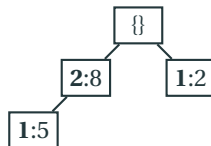
## Review Question 2 – FP-Tree

Considering the same transactions above, use frequent-pattern (FP) growth algorithm to perform frequent itemsets mining with a support threshold of 30%.

1. What is the item head table? (Hint: keep in mind the support count and sort order.)
2. What is the FP-tree corresponding to the transactions above?

| Item | $\sigma$ | Node Link |
|:----:|:--------:|:----------|
| **2** | 8 | |
| **1** | 7 | |
| **3** | 6 | |
| **5** | 3 | |

**Table 4:** Header Table



*Node Links* are omitted. They are pointers which point to the node with the corresponding TID.
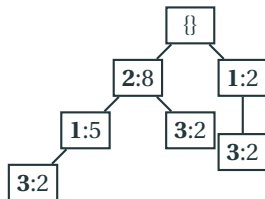
## Review Question 2 – FP-Tree

Considering the same transactions above, use frequent-pattern (FP) growth algorithm to perform frequent itemsets mining with a support threshold of 30%.

1. What is the item head table? (Hint: keep in mind the support count and sort order.)
2. What is the FP-tree corresponding to the transactions above?

| Item | $\sigma$ | Node Link |
|------|----------|-----------|
| **2** | 8 | |
| **1** | 7 | |
| **3** | 6 | |
| **5** | 3 | |

**Table 4:** Header Table



*Node Links* are omitted. They are pointers which point to the node with the corresponding TID.
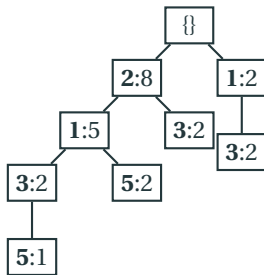
## Review Question 2 – FP-Tree

Considering the same transactions above, use frequent-pattern (FP) growth algorithm to perform frequent itemsets mining with a support threshold of 30%.

1. What is the item head table? (Hint: keep in mind the support count and sort order.)
2. What is the FP-tree corresponding to the transactions above?

| Item | $\sigma$ | Node Link |
|------|----------|-----------|
| **2** | 8 | |
| **1** | 7 | |
| **3** | 6 | |
| **5** | 3 | |

**Table 4:** Header Table

*Node Links* are omitted. They are pointers which point to the node with the corresponding TID.

## Review Question 2 – FP-Tree
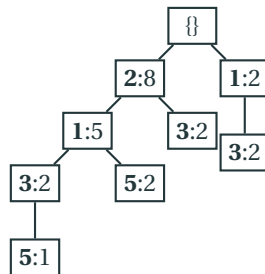
What is the conditional pattern base corresponding to the transactions above?

| Item | $\sigma$ | Node Link |
|------|----------|-----------|
| **2** | 8 | |
| **1** | 7 | |
| **3** | 6 | |
| **5** | 3 | |

**Table 5:** Header Table



| Item | Conditional Pattern Base |
|------|--------------------------|
| **5** | {(**1**:2, **2**:2), (**3**:1, **1**:1, **2**:1) } |
| **3** | {(**1**:2, **2**:2), (**2**:2), (**1**:2) } |
| **1** | {(**2**:5) } |
| **2** | {} |

- anti-monotone holds true.
- For each item, the sum of frequency count in all nodes is equal to $\sigma(X)$.
- In conditional pattern base, items in the same itemset should have the same frequency count.

34