Q-Learning

Objectives

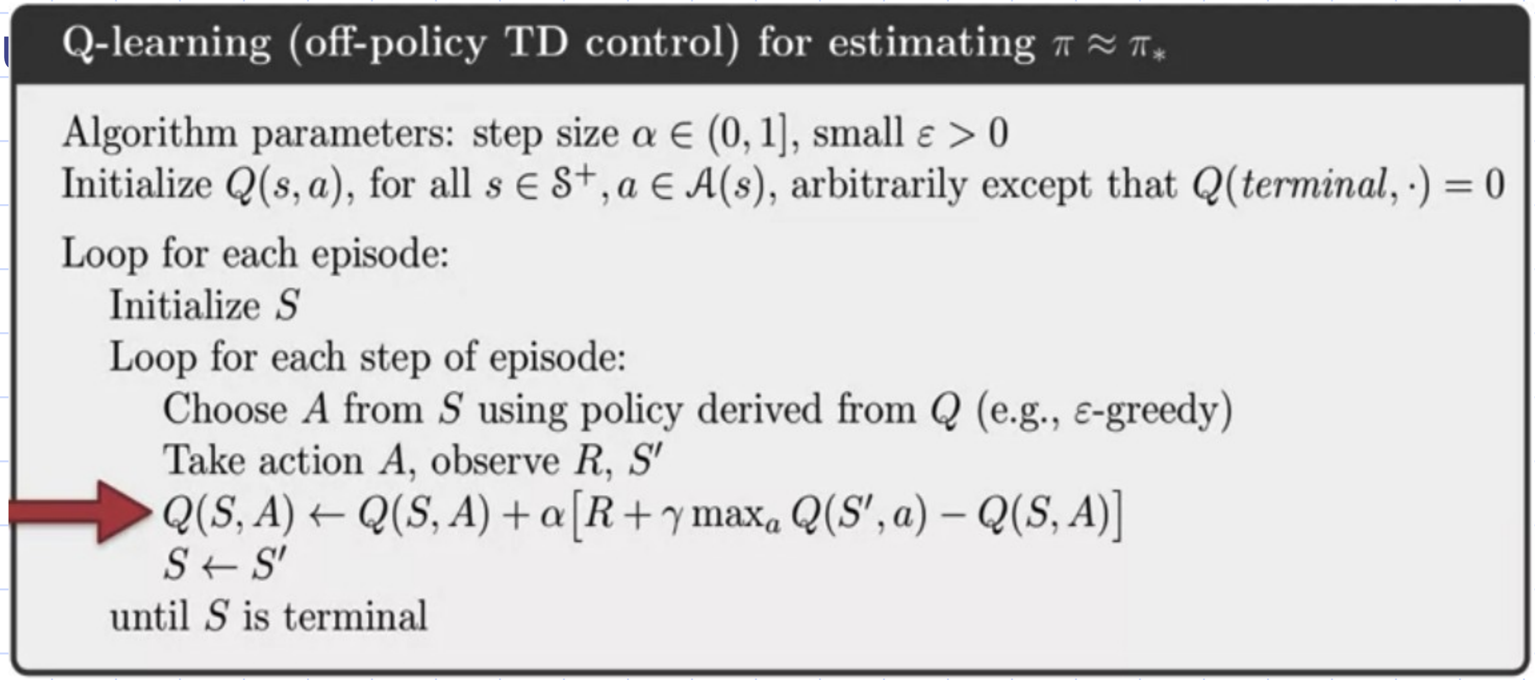
- ❑ Describe the Q-learning algorithm
- ❑ Apply Q-learning to an MDP to find the optimal policy
- ❑ Understand how Q-learning performs in an example
- ❑ Understand the differences between Q-learning and Sarsa

Q-Learning

- ❑ Q-Learning is used to learn the optimal action-value function Q∗(s,a), which represents the expected cumulative reward starting from state s and taking action a, under an optimal policy.
- ❑ Q-Learning is an off-policy algorithm (learns from data generated by an exploratory policy while simultaneously estimating the value of the optimal policy)

Q-Learning

□ The new element in Q-learning is the action value
u



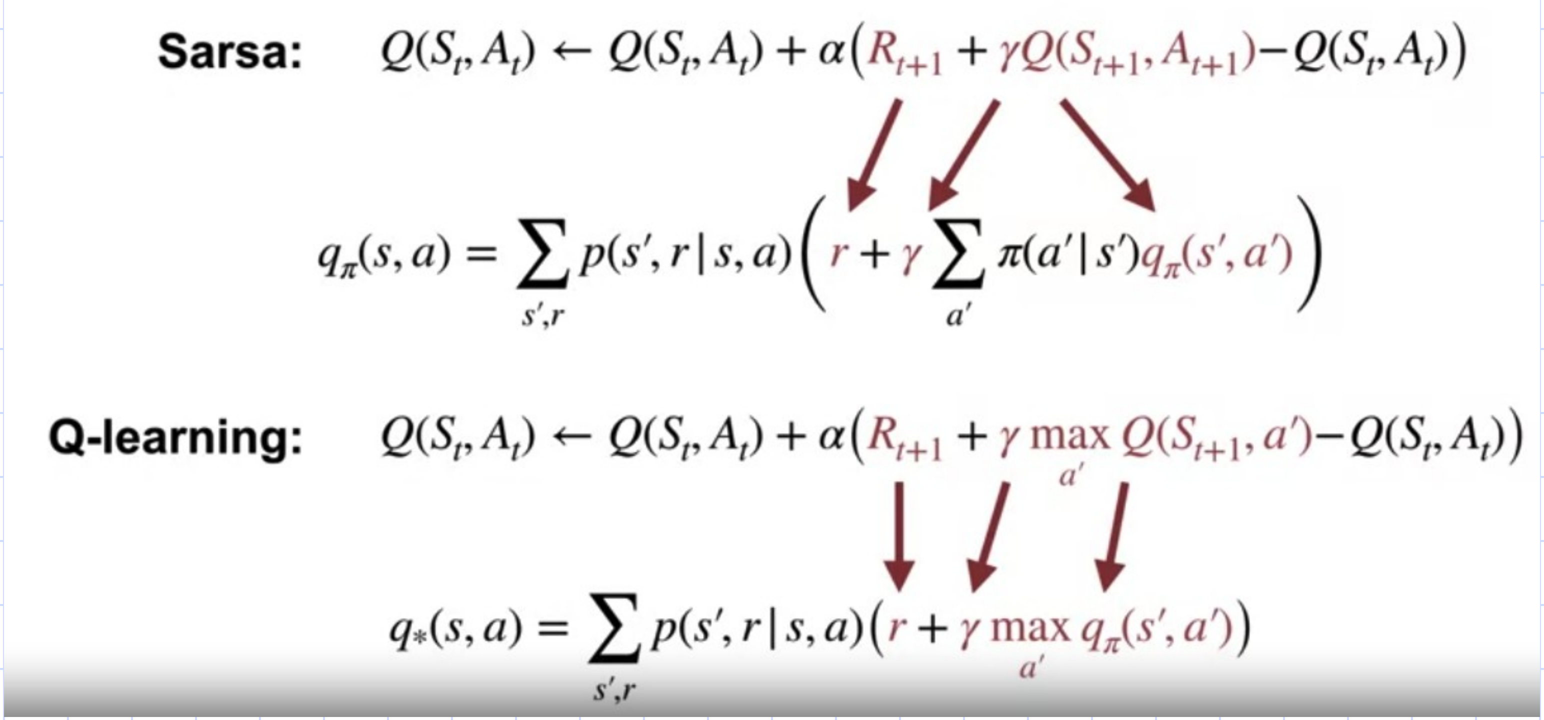
```
Q-learning (off-policy TD control) for estimating $\pi \approx \pi_*$

Algorithm parameters: step size $\alpha \in (0,1]$, small $\varepsilon > 0$
Initialize $Q(s,a)$, for all $s \in \mathcal{S}^+, a \in \mathcal{A}(s)$, arbitrarily except that $Q(terminal,\cdot) = 0$

Loop for each episode:
    Initialize $S$
    Loop for each step of episode:
        Choose $A$ from $S$ using policy derived from $Q$ (e.g., $\varepsilon$-greedy)
        Take action $A$, observe $R$, $S'$
        $Q(S,A) \leftarrow Q(S,A) + \alpha\big[R + \gamma \max_a Q(S',a) - Q(S,A)\big]$
        $S \leftarrow S'$
    until $S$ is terminal
```

Q-Learning

- Q-learning solves the Bellman equation using samples from the environment.
- Q-learning uses the Bellman's Optimality Equation for action values.
- The optimality equations enable Q-learning to directly learn Q-star instead of switching between policy improvement and policy evaluation steps.

Q-Learning

- □ Revisiting Bellman equations



Sarsa: $Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha\big(R_{t+1} + \gamma Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t)\big)$

$$q_\pi(s, a) = \sum_{s',r} p(s', r \mid s, a)\Big(r + \gamma \sum_{a'} \pi(a' \mid s') q_\pi(s', a')\Big)$$

Q-learning: $Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha\big(R_{t+1} + \gamma \max_{a'} Q(S_{t+1}, a') - Q(S_t, A_t)\big)$

$$q_*(s, a) = \sum_{s',r} p(s', r \mid s, a)\big(r + \gamma \max_{a'} q_\pi(s', a')\big)$$

Q-Learning

- Q-Learning learns the optimal action-value function by iteratively updating Q-values based on observed rewards and transitions.
- It is widely used in various domains, including robotics, game playing, and autonomous systems, for learning optimal decision-making policies in complex environments.

Q-Learning vs SARSA

- SARSA and Q-Learning differ in their approach to policy updates, action selection, and convergence properties.
- SARSA learns under the current policy being evaluated,
- Q-Learning learns under an exploratory policy while simultaneously estimating the optimal policy.

Q-Learning vs SARSA

- ❑ Policy Type:
  - ❑ SARSA is an on-policy learning algorithm. It updates its Q-values based on the action chosen by the current policy.
  - ❑ Q-Learning: Q-Learning is an off-policy learning algorithm. It updates its Q-values based on the action that maximizes the Q-value of the next state.

Q-Learning vs SARSA

- ❑ Update Rule:
  - ❑ SARSA: SARSA uses the SARSA update rule, which updates the Q-values based on the current action and the action chosen by the policy in the next state:

$$Q(s,a) \leftarrow Q(s,a) + \alpha\left(r + \gamma Q(s',a') - Q(s,a)\right)$$

  - ❑ Q-Learning: Q-Learning uses the Q-Learning update rule, which updates the Q-values based on the maximum Q-value of the next state:

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

Q-Learning vs SARSA

- Action Selection:
  - SARSA: SARSA selects actions based on the current policy being evaluated.
  - Q-Learning: Q-Learning selects actions based on an exploratory policy.

Q-Learning vs SARSA

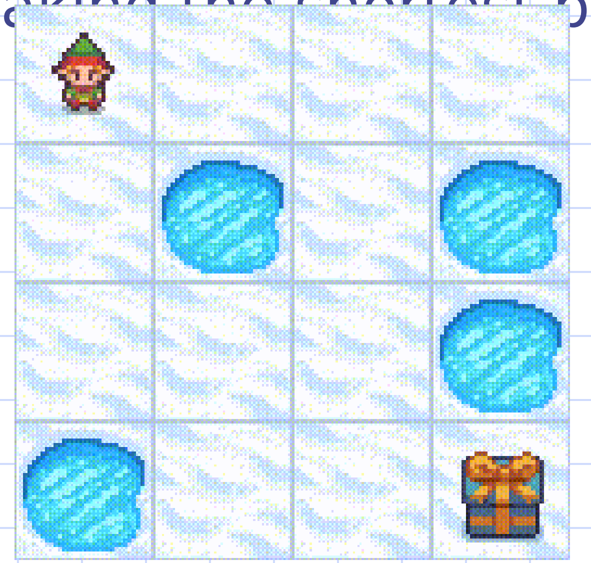
- Convergence:
  - SARSA: SARSA converges to a policy that is ε-optimal under the current policy being evaluated.
  - Q-Learning: Q-Learning converges to the optimal action-value function $Q*(s,a)$ and subsequently the optimal policy $\pi*(s)$.

Q-Learning vs SARSA

- Safety:
  - SARSA: SARSA is generally safer to use in environments where exploration may lead to dangerous actions, as it learns under the current policy.
  - Q-Learning: Q-Learning may take riskier actions during exploration, as it learns under an exploratory policy while simultaneously estimating the optimal policy.

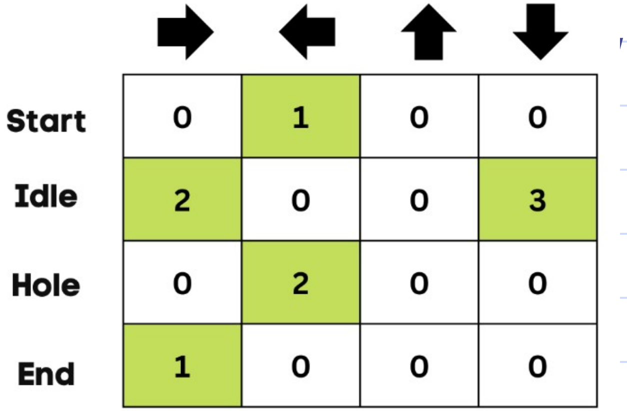
Q-Learning- Example

- ❑ In the frozen lake environment, the agent must cross the frozen lake from the start to the goal, without falling into the holes. The best strategy is to reach goals by taking the shortest path.



Q-Learning- Example

- ❑ Q- Table:
  - ❑ The agent utilizes a Q-table to determine the optimal action to take based on the expected reward associated with each state in the environment. In essence, a Q-table serves as a structured representation of actions and states, and the Q-learning algorithm is employed to continually refine ithin this table.



	→	←	↑	↓
Start	0	1	0	0
Idle	2	0	0	3
Hole	0	2	0	0
End	1	0	0	0

Q-Learning- Example

- Q-Learning Algorithm:
  - Step 1: Initialize Q- Table
  - Step 2: Choose an Action
  - Step 3: Perform Action
  - Step 4: Measure Reward
  - Step 5: Update Q- Table: after multiple iteration a good Q-Learing is ready
  - Step 6: back step 2

Summary

- ❑ Describe the Q-learning algorithm
- ❑ Apply Q-learning to an MDP to find the optimal policy
- ❑ Understand how Q-learning performs in an example MDP
- ❑ Understand the differences between Q-learning and Sarsa

Q & A