John Smith, Ph.D.

Department of Computer Science
University Name

Email: email@university.edu
Website: www.university.edu

July 13, 2025

Introduction to Q-learning and SARSA

Overview

In this chapter, we will explore two fundamental algorithms in Reinforcement Learning: **Q-learning** and **SARSA**. These algorithms are critical for learning optimal policies to maximize cumulative rewards in an environment.

Key Concepts - Part 1

- **Reinforcement Learning Basics**:
  - An agent interacts with an environment through actions.
  - Receives feedback in forms of rewards to update its knowledge.
  - Objective: Learn a policy that maximizes expected rewards.
- **Q-learning**:
  - A model-free, off-policy algorithm.
  - Learns the value of actions in a state, known as the **Q-value**.
- **SARSA**:
  - An on-policy algorithm that updates Q-values based on the agent's actions.

Key Comparisons

Feature	Q-learning	SARSA
Policy Type	Off-policy	On-policy
Update Rule	Uses maximum Q-value	Uses the action taken
Exploration Strategy	More flexible	More conservative
Convergence	Guarantees under certain conditions	Converges at the current policy

Table: Comparison of Q-learning and SARSA

Mathematical Notation

- **Q-value Update Equation (Q-learning)**:

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left(r + \gamma \max_{a'} Q(s',a') - Q(s,a) \right) \tag{1}$$

- **Q-value Update Equation (SARSA)**:

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left(r + \gamma Q(s',a') - Q(s,a) \right) \tag{2}$$

- Where:
  - s = current state
  - a = action taken
  - r = reward received
  - s' = next state
  - a' = action taken in next state
  - α = learning rate
  - γ = discount factor

Overview of Q-learning and SARSA

Key Concepts

Introduction to Reinforcement Learning (RL) strategies, focusing on Q-learning and SARSA as foundational algorithms.

Q-learning

- **Definition**: Q-learning is an off-policy method that learns the value of the optimal policy regardless of the actions taken.
- **Q-value Update Rule**:

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left(R + \gamma \max_{a} Q(s',a') - Q(s,a) \right) \tag{3}$$

  - s: current state
  - a: action taken
  - R: reward received after taking action a
  - s': next state after taking action a
  - γ: discount factor
  - α: learning rate
- **Example**: In a grid world, reaching a goal state might update the Q-value to reflect a reward of $+10$.

SARSA: State-Action-Reward-State-Action

- **Definition**: SARSA is an on-policy method that updates Q-values based on the agent's actual actions, learning from the current policy.
- **Q-value Update Rule**:

$$Q(s, a) \leftarrow Q(s, a) + \alpha \left(R + \gamma Q(s', a') - Q(s, a)\right) \tag{4}$$

  - a': next action taken in state s'
- **Example**: Different actions taken in the grid world will affect Q-values, potentially leading to slower convergence.

Comparison of Q-learning and SARSA

- **Learning Type**:
  - Q-learning: Off-policy
  - SARSA: On-policy
- **Exploration vs. Exploitation**: Both implement epsilon-greedy strategies.

Conclusion - Chapter 3: Q-learning and SARSA

1. Summary of reinforcement learning algorithms focusing on Q-learning and SARSA.
2. Importance of understanding off-policy and on-policy methods.
3. Practical implications and applications in various domains.

Summary of Key Concepts

Q-learning

- Off-policy reinforcement learning algorithm.
- Learns optimal action value without any environmental model.
- Q-value update:

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

Summary of Key Concepts (continued)

SARSA (State-Action-Reward-State-Action)

- On-policy reinforcement learning algorithm.
- Q-value updates based on the action taken in the next state:

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

Comparison

- Q-learning: off-policy, learns optimal policy independently of agent's behavior.
- SARSA: on-policy, learns from actions taken by current policy.

Practical Implications and Key Points

- Q-learning more effective in exploratory contexts.
- SARSA preferred in safety-critical environments.
- Both methods focus on maximizing cumulative rewards through interaction.
- Understanding exploration vs. exploitation is critical for algorithm effectiveness.

Application Example

Game Playing

- Q-learning: Suitable for complex games allowing aggressive exploration (e.g., chess).
- SARSA: Better for scenarios where safety matters (e.g., human-robot interactions).