Lab6: Q Learning

August 2023

1 In This Lab

1.1 Topics to Cover

- Q learning

1.2 Requirements

- Python installed on your computer.
- Basic Python programming, Jupyter Notebook.
- Libraries: gym, scikit-learn, numpy, matplotlib, seaborn

1.3 Environment

- OpenAI Gym

2 Q-Learning

Q-learning is a reinforcement learning algorithm that aims to learn an optimal action-selection policy for an agent in an environment. The agent interacts with the environment by taking actions, receiving rewards, and learning to maximize its cumulative rewards over time.

2.1 Key Concepts

- **State (s):** A representation of the agent's current situation in the environment.
- **Action (a):** A decision made by the agent to transition from one state to another.
- **Action-Value Function (Q(s, a)):** The expected cumulative reward an agent can achieve by taking action "a" in state "s" and following the optimal policy thereafter.

- **Bellman Equation:** A recursive equation that relates the action-value function of a state-action pair to the immediate reward and the maximum expected future reward from the next state.
- **Exploration vs. Exploitation:** Balancing between exploring new actions to discover their effects and exploiting known actions to maximize rewards.

2.2 Algorithm Steps

1. Initialize the action-value function Q(s, a) arbitrarily for all state-action pairs.
2. Repeat the following steps until convergence:
   (a) Select an action based on a policy (often epsilon-greedy) that balances exploration and exploitation.
   (b) Execute the chosen action in the environment and observe the reward and the next state.
   (c) Update the action-value function using the Bellman equation:

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

   where α is the learning rate, r is the observed reward, γ is the discount factor, and s' is the next state.

2.3 Convergence and Optimality

Q-learning converges to the optimal action-value function $Q^*(s,a)$ under certain conditions, assuming that all state-action pairs are visited infinitely often. The optimal policy can be derived from $Q^*(s,a)$ by selecting the action with the highest Q-value for each state.

2.4 Task 1:

Follow the Q learning grid world example (provided)

2.5 Report

Use a different environment from the gym and follow the same steps as in the grid world example.