Introduction to Reinforcement Learning Course - Home Assignment 1

Dr. Teddy Lazebnik

1.10.2024

Abstract

This document outlines the first home assignment for an introductory Reinforcement Learning course. The assignment aims to introduce students to fundamental RL concepts through practical exercises. Students are tasked with defining a Markov Decision Process (MDP) for a grid-world environment, implementing the value iteration algorithm, and formulating a real-world problem as an RL problem.

Objective

The goal of this assignment is to familiarize yourself with the fundamental concepts of Reinforcement Learning (RL) and to start thinking about RL problems.

Task 1 - theory (30% of the score)

Question 1: Prove that the policy iteration algorithm converges to the optimal policy for a finite MDP with a finite action space for each state.

Question 2: Prove that the Bellman optimality equation has a unique solution for a finite MDP with a bounded reward function.

Question 3: Analyze the challenges of using linear function approximation in value-based reinforcement learning. Discuss the potential issues that may arise and propose methods to mitigate them.

Task 2 - practice (70% of the score)

Consider a simple grid world environment with a starting state, a goal state, and obstacles. The agent can move up, down, left, or right. The agent receives a reward of +1 upon reaching the goal state and a reward of -1 for hitting an obstacle. The agent's goal is to find a policy that maximizes the expected cumulative reward.

- Define the MDP: Clearly define the states, actions, transition probabilities, and reward function for this grid world environment. In more detail: Clearly specify the states: Each grid cell represents a state. Define the actions: The agent can move up, down, left, or right. Determine the transition probabilities: For each state and action, specify the probability of landing in each possible next state. Consider factors like obstacles and edge cases (e.g., moving off the grid). Define the reward function: Assign rewards for reaching the goal, hitting obstacles, and taking any other action.

- Value Iteration: Implement the value iteration algorithm to compute the optimal value function for this grid world. In more detail: Write code to implement the value iteration algorithm. Initialize the value function for all states. Iteratively update the value function using the Bellman optimality equation until convergence. Use the right discount factor - for instance ($\gamma = 0.9$) as an initial guess.

- Policy Extraction: Extract the optimal policy from the computed value function. In more detail: Once the value function converges, determine the optimal action for each state based on the maximum expected future reward.

- Visualization: Visualize the grid world, the value function, and the optimal policy. In more detail: Create a visual representation of the grid world. Color-code the grid cells based on their value function. Indicate the optimal policy for each state using arrows or other symbols.

To be exact, students should submit a PDF report, Python/JS code, and visualization. For the PDF report, the report should include:

- A detailed description of the grid world MDP, including states, actions, transition probabilities, and the reward function.
- A step-by-step explanation of the value iteration algorithm implementation.
- A clear description of how the optimal policy was extracted from the value function.

In addition, the code used to implement the value iteration algorithm. This should be included as a ZIP file with the code, readme file, and requirements.txt with all the required library installs.

For the visualizations: 1) a clear and informative visualization of the grid world, including obstacles, starting state, and goal state; 2) A visualization of the computed value function for each state; and 3) A visualization of the optimal policy, indicating the best action for each state.