National Taiwan University
Department of Electrical Engineering
Name: Tzu-Tsen, Hsieh
ID: R11921072

113-1 (Fall 2024)
Reinforcement Learning

Assignment 1 Report

Q1. What methods have you tried for async DP? Compare their performance.

1 In-place Dynamic Programming

- In-place dynamic programming updates the value of each state directly during each iteration, which reduces memory usage since it doesn't need to store both old and new values. This method improves efficiency by allowing immediate reflection of changes in state values, speeding up convergence. However, it requires careful management of update order to ensure accurate calculations, as some states may rely on partially updated values of others.
- The experimental result is: 1056 steps.

2 Prioritized Sweeping

- It evaluates the maximum Bellman error to prioritize state convergence, allowing it to skip unnecessary steps for states that have already converged or are close to convergence. By focusing on states with the highest errors, this method efficiently directs computational resources to areas needing updates, reducing iterations and improving overall efficiency. This targeted approach helps prevent unnecessary recalculations and allows for faster convergence, particularly in large state spaces where some states stabilize more quickly than others.
- The experimental result is: 1004 steps.

3 Real-time dynamic programming (RTDP)

- Updating the state based on the agent's movement has a significant drawback: certain states may never be visited by the agent, resulting in many iterations before reaching the optimal solution. While it can achieve **partial** optimality in the early stages, the policy remains suboptimal globally. This is especially true if the agent does not pass through some critical states during its path.
- The experimental result is: 2007 steps.

4 Comparison

- In my experience, the fastest method to converge is Prioritized Sweeping because it focuses on the largest changes in state values. By prioritizing states with the highest Bellman errors, this approach accelerates convergence by applying updates where they can have the most significant impact. This targeted focus reduces the number of unnecessary iterations, making the learning process more efficient, especially in complex environments with many states.
- Steps count: Prioritized Sweeping > In-place Dynamic Programming > RTDP

Q2. What is your final method? How is it better than other methods you've tried?

I choose prioritized sweeping as my final result. Because it optimizes the update process by focusing on the most important states first. It will be more efficient to have fewer steps to converge to the optimal policy. Besides, it also combines the benefit of in-place dynamic programming which replace the state at the same time.

1 Sync DP result:

- Iterative Policy Evaluation: Solved in 7568 steps
- Policy Iteration: Solved in 3256 steps
- Value Iteration: Solved in 1144 steps

2 Async DP result:

- In-place Dynamic Programming: Solved in 1056 steps
- Prioritized Sweeping: Solved in 1004 steps
- Real-time dynamic programming (RTDP): Solved in 2007 steps

3 Novel Method

The novel method I have tried combines Real-Time Dynamic Programming (RTDP) with Prioritized Sweeping. Initially, Bellman error is computed for each state and stores the state with the largest error in a priority queue. The main loop then continuously selects the state with the highest priority (i.e., the largest Bellman error) from the queue and updates its value. It recalculates the Bellman error for each state after every update and pushes states with significant errors back into the queue. This process integrates real-time action selection and state transitions, allowing for efficient value updates while progressively improving the policy towards optimality.

Solved in 1460 steps