COMP-579: Reinforcement Learning - Assignment 2

Posted Tuesday, February 4, 2025
Due Friday, February 21, 2025

The assignment contains a coding section and a math section. Please submit a PDF/IPYNB file containing your solutions to the problems, and the code you used to produce the results.

1. **Coding: Tabular RL [70 points]**

   In this problem, compare the performance of SARSA and expected SARSA on the Frozen Lake Problem from the Gym environment suite:

   `https://gymnasium.farama.org/environments/toy_text/frozen_lake/`

   You can choose to start with the template notebook below. If you are not familiar with the Gym API, play around with it first.

   `https://colab.research.google.com/drive/1CFXs0r4MncB-iSfL2OA8dZbA3cVs-jd8#scrollTo=OevfjgjL8mFk&uniqifier=1`

   Using a **tabular** representation of the state space, test different temperatures and learning rates, and measure the return overtime. The agent's exploration should be softmax (Boltzmann).

   For each experiment (hyperparamter combination), do 10 independent trials. Each trial consists of 500 segments. In each segment, there are 10 episodes of training, followed by 1 testing episode where you run the optimal policy so far (i.e. pick actions greedily). In other words, there are 5500 episodes in each trial. Pick at least 3 settings of the temperature parameter used in the exploration and at least 3 settings of the learning rate. Plot:

   - Three graph that shows the effect of the parameters on the final ***training*** performance.

     1. In one graph, the x-axis shows the alpha (learning rate), and the y-axis shows the return of the agent (averaged over the last 10 training episodes and the 10 runs). The graph should have at least 3 lines (for 3 temperature values).

     2. In the second graph,the x-axis shows the temperature, and the y-axis shows the return of the agent (averaged over the last 10 training episodes and the 10 runs). the graph should have at least 3 lines (for 3 learning rates)

     3. In the third graph, show the training reward as a function of the number of episodes.

     There should be 9 lines in total, one for each of temperature/alpha combinations

   - **4.** The graph that instead shows the effect of the parameters on the final ***testing*** performance. The y-axis should now show the test reward(greedy policy) as a function of the number of the episodes. (There should be 9 lines in total, one for each of temperature/alpha combinations)

   Feel free to create other figures to help you analyze the results. Write a small report that describes your experiment, your choices of parameters, and the conclusions you draw from the graphs.

MDP Description

Let the MDP have three states $S = \{s_1, s_2, s_3\}$, where s_3 is the terminal state. The actions available in s_1 and s_2 are $A = \{a_1, a_2\}$. Transition probabilities and rewards are as follows:

- **State s_1:**
  - $P(s_2|s_1, a_1) = 1$, reward $R(s_1, a_1) = 2$.
  - $P(s_3|s_1, a_2) = 1$, reward $R(s_1, a_2) = 5$.
- **State s_2:**
  - $P(s_3|s_2, a_1) = 1$, reward $R(s_2, a_1) = 1$.
  - $P(s_1|s_2, a_2) = 1$, reward $R(s_2, a_2) = -1$.
- **State s_3:** Terminal state with $v(s_3) = 0$.

The discount factor is $\gamma = 0.9$.

2: Policy Evaluation (Given Policy π)

Define the following **deterministic policy π:**

- $\pi(s_1) = a_1$.
- $\pi(s_2) = a_1$.

Tasks:

(a) **Solve the Bellman equations for v_π analytically.**
Derive the value function v_π for the policy π by solving the Bellman equations using matrix inversion.

(b) **Solve v_π numerically through iterative updates.**
Use iterative updates to compute v_π until convergence, given the transition probabilities.

3: Finding the Optimal Policy

Solve the Bellman optimality equations for the MDP described above.

Tasks:

(a) **Find the optimal value function v_* analytically.**
Derive the Bellman optimality equations for this MDP and solve them to find $v_*(s_1)$, $v_*(s_2)$, $v_*(s_3)$.

(b) **Find v_* numerically using value iteration.**
Perform value iteration to compute the optimal value function v_* and identify the optimal policy π_*.

Please share the Python code and the corresponding results you used for the calculations(For Q2 and Q3).