COMPSCI 687 Homework 5
(100 Points)

Instructions: Submit your answers as a single typed PDF file in Gradescope. You may use any software to create the PDF, and any reasonable formatting of your answers will suffice. You may collaborate with other students and may use generative AI to complete this assignment without reporting the collaboration or generative AI usage, but you are responsible for the accuracy of your answers.

Part One: Written (60 Points)

1. (40 Points) In this problem you will prove that Dynamic Programming Policy Evaluation converges to v^π. Assume that the MDP is finite, rewards are bounded, and $\gamma \in [0, 1)$. Let $v_0 : \mathcal{S} \to \mathbb{R}$ be arbitrary.

   (a) Write the definition of an operator $T^\pi : \mathbb{R}^{|\mathcal{S}|} \to \mathbb{R}^{|\mathcal{S}|}$ that represents one full, synchronous Dynamic Programming Policy Evaluation update. Write your definition with the left side in the form

   $$\forall s \in \mathcal{S},\, T^\pi v(s) = \cdots$$

   so that $v_{i+1} = T^\pi v_i$.

   (b) Show that v^π is a fixed point of T^π. **Hint:** Compare the equation $T^\pi v^\pi = v^\pi$ with the Bellman equation for v^π. **Note:** The answer can be quite short; a single sentence is enough. This only shows that v^π is a fixed point. It does not show that the updates converge to this fixed point.

   (c) Prove that T^π is a contraction mapping under the max norm,

   $$\|v - v'\|_\infty = \max_{s \in \mathcal{S}} |v(s) - v'(s)|.$$

   State the contraction rate.

   (d) Use the Banach fixed-point theorem to prove that the sequence $v_{i+1} = T^\pi v_i$ converges to v^π. State why the theorem applies. **Note:** This part might be brief; it should bring together the results that you have shown above.

2. (20 Points) Construct a finite episodic MDP and policy for which a one-episode every-visit Monte Carlo estimate of $v^\pi(s)$ is biased. Assume that the episode starts in s, and let $t_1 < \cdots < t_N$ be all of the times that s is visited in that episode. The estimate is

   $$\widehat{v}_{\text{EV}}(s) = \frac{1}{N} \sum_{i=1}^{N} G_{t_i}.$$

   Specify all components of your MDP and the policy. Compute $v^\pi(s)$ and $\mathbf{E}[\widehat{v}_{\text{EV}}(s)]$, and show that they are not equal. This is an average over all visits within one episode, not an average over visits collected from multiple episodes. Your example must have positive probability of visiting s more than once.

Part Two: Programming (40 Points)

3. (40 Points) Implement Value Iteration for the state-value function in any programming language. Your program should read an MDP from a file and write the resulting optimal value function and the optimal action or actions for each state to two output files. Use the MDP file format from Homework 4, repeated below. All numbers are separated by spaces, and there are no headers or comments.

   MDP file: States are numbered $0, \ldots, n-1$ and actions are numbered $0, \ldots, m-1$. If the MDP has a terminal absorbing state, s_∞, it is included in these n states. The file contains:

   - Line 1: n and m, the numbers of states and actions.
   - Line 2: γ.
   - Line 3: $d_0(0), \ldots, d_0(n-1)$, written as n numbers separated by spaces.
   - The remaining nm lines: one line for each state-action pair, ordered first by state and then by action, starting with $(0,0), (0,1), \ldots, (0,m-1), (1,0), \ldots$. Each line contains $n+3$ numbers:

   $$s \quad a \quad R(s,a) \quad p(s,a,0) \quad p(s,a,1) \quad \cdots \quad p(s,a,n-1).$$

   Value output file: Write n lines, with one number per line: $\widehat{v}^*(0), \ldots, \widehat{v}^*(n-1)$, in that order. Write at least eight digits after the decimal point.

   Optimal-action output file: Write n lines. Line $s+1$ contains the numbers of all optimal actions for state s, in increasing order and separated by spaces. If only one action is optimal, the line contains one number. If multiple actions are optimal, output all of them.

   Use double precision, initialize every state value to zero, and use the synchronous Value Iteration update from class: each complete update uses only values from the previous iteration. Stop when the maximum absolute change over all states in one complete update is at most 10^{-10}, and output the newly computed vector. Compute

   $$\widehat{q}(s,a) = \sum_{s' \in \mathcal{S}} p(s,a,s') \left(R(s,a) + \gamma \widehat{v}^*(s') \right) \quad \text{and} \quad M_s = \max_{a \in \mathcal{A}} \widehat{q}(s,a).$$

   For the supplied files, output every action a satisfying $M_s - \widehat{q}(s,a) \leq 10^{-8}$.

   First, test your program using `687_gridworld_MDP.txt`. The files `687_Gridworld_opt_values.txt` and `687_Gridworld_opt_actions.txt` contain the correct outputs. **There may be slight numerical differences** due to rounding and stopping tolerances; your values should agree with the reference to within 2×10^{-6} at every state, and your optimal-action output should match the reference. The notes' gridworld locations $1, \ldots, 23$ correspond to file states $0, \ldots, 22$, and the terminal absorbing state is file state 23. Actions $0, 1, 2, 3$ are AU, AD, AL, AR, respectively.

   Then run your program using `test_MDP.txt`. This MDP has seven states and five actions. Some states have one optimal action and some have multiple optimal actions. In your PDF answer, report your estimates of $v^*(0), \ldots, v^*(6)$ and all optimal actions for each state. Give at least six digits after the decimal point for the values. You do not need to submit your source code or output files, or report the gridworld values or actions.