COMPSCI 687 Homework 4
(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)

Note: For this part, assume that we're considering only finite MDPs with bounded rewards and $\gamma < 1$. You may use results proved in the course notes, but should reference any such results that you use (e.g., "By Property 3 from the course notes...").

1. (8 Points) Derive the Bellman equation for q^π. Begin with the definition of $q^\pi(s,a)$ as an expected discounted return, and end with a recursive expression for $q^\pi(s,a)$ that uses only $R, p, \gamma, \pi, \mathcal{S}, \mathcal{A}$, and q^π. Your final answer should not contain v^π, expectations, or random variables. Show your work and explain the steps.

   We begin with the definition of $q^\pi(s,a)$:

$$q^\pi(s,a) = \mathbf{E}\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k} \middle| S_t = s, A_t = a; \pi\right] \tag{1}$$

$$\stackrel{(a)}{=} \mathbf{E}\left[R_t + \gamma \sum_{k=0}^{\infty} \gamma^k R_{t+1+k} \middle| S_t = s, A_t = a; \pi\right] \tag{2}$$

$$\stackrel{(b)}{=} R(s,a) + \gamma \mathbf{E}\left[\sum_{k=0}^{\infty} \gamma^k R_{t+1+k} \middle| S_t = s, A_t = a; \pi\right] \tag{3}$$

$$\stackrel{(c)}{=} R(s,a) + \gamma \sum_{s' \in \mathcal{S}} p(s,a,s') \sum_{a' \in \mathcal{A}} \pi(s',a') \mathbf{E}\left[\sum_{k=0}^{\infty} \gamma^k R_{t+1+k} \middle| S_{t+1} = s', A_{t+1} = a'; \pi\right] \tag{4}$$

$$\stackrel{(d)}{=} R(s,a) + \gamma \sum_{s' \in \mathcal{S}} p(s,a,s') \sum_{a' \in \mathcal{A}} \pi(s',a') q^\pi(s',a'). \tag{5}$$

   Here **(a)** separates the first reward and reindexes the remaining sum, **(b)** uses linearity of expectation and the definition of $R(s,a)$, **(c)** uses the law of total expectation, the definitions of p and π, and the Markov property, and **(d)** uses the definition of q^π.

2. (8 Points) Before looking up the answer, write what you think the Bellman optimality equation for q^* is, using the notation of this course. You may guess or derive it; a derivation is not required. Then look up the equation in the first edition of Sutton and Barto's *Reinforcement Learning: An Introduction* at `http://incompleteideas.net/book/first/ebook/node35.html`. Write the correct equation in our notation. You may use AI to help translate the notation. State whether your original expression was correct. If it was not, explain the difference. Include both your original expression and the correct equation, even if they agree.

   Notation: Their Q^* is our q^*, and their $\mathcal{P}^a_{ss'}$ is our $p(s,a,s')$. Their $\mathcal{R}^a_{ss'}$ is the expected reward conditioned on the next state being s', so our reward function is $R(s,a) = \sum_{s' \in \mathcal{S}} p(s,a,s') \mathcal{R}^a_{ss'}$.

$$q^*(s,a) = R(s,a) + \gamma \sum_{s' \in \mathcal{S}} p(s,a,s') \max_{a' \in \mathcal{A}} q^*(s',a'). \tag{6}$$

3. (14 Points) Let π be a policy such that q^π satisfies the Bellman optimality equation from Question 2. Prove that $q^\pi = q^*$.

   We follow the unrolling approach from the proof of Property 6 in the course notes. To show that $q^\pi = q^*$, we will show that for every policy μ and state-action pair (s,a), $q^\pi(s,a) \geq q^\mu(s,a)$. From the definition of $q^*(s,a)$ it is clear that $q^\pi(s,a)$ cannot be greater than $q^*(s,a)$, and so this allows us to conclude that $q^\pi(s,a) = q^*(s,a)$.

   Let μ be any policy. For every state-action pair (s,a),

$$\begin{aligned}
q^\pi(s,a) &\overset{(a)}{=} \sum_{s' \in \mathcal{S}} p(s,a,s') \Big(R(s,a) + \gamma \max_{a' \in \mathcal{A}} q^\pi(s',a') \Big) && (7)\\
&\overset{(b)}{\geq} \sum_{s' \in \mathcal{S}} p(s,a,s') \Big(R(s,a) + \gamma \sum_{a' \in \mathcal{A}} \mu(s',a') q^\pi(s',a') \Big) && (8)\\
&\overset{(c)}{=} \sum_{s' \in \mathcal{S}} p(s,a,s') \Big(R(s,a) && (9)\\
&\quad + \gamma \sum_{a' \in \mathcal{A}} \mu(s',a') \sum_{s'' \in \mathcal{S}} p(s',a',s'') \Big(R(s',a') + \gamma \max_{a'' \in \mathcal{A}} q^\pi(s'',a'') \Big) \Big) && (10)\\
&\overset{(d)}{\geq} \sum_{s' \in \mathcal{S}} p(s,a,s') \Big(R(s,a) && (11)\\
&\quad + \gamma \sum_{a' \in \mathcal{A}} \mu(s',a') \sum_{s'' \in \mathcal{S}} p(s',a',s'') \Big(R(s',a') + \gamma \sum_{a'' \in \mathcal{A}} \mu(s'',a'') q^\pi(s'',a'') \Big) \Big). && (12)
\end{aligned}$$

   Here **(a)** follows from the assumption that q^π satisfies the Bellman optimality equation, **(b)** follows because the average over actions under any distribution (here $\mu(s',\cdot)$) is at most the maximum over actions, **(c)** follows by replacing $q^\pi(s',a')$ with its Bellman optimality equation, and **(d)** follows because the average over actions under $\mu(s'',\cdot)$ is at most the maximum over actions.

   If we continue this process of expanding the expression using the Bellman optimality equation, then after n expansions we have that

$$q^\pi(s,a) \geq \mathbf{E} \left[\sum_{k=0}^{n-1} \gamma^k R_{t+k} + \gamma^n q^\pi(S_{t+n}, A_{t+n}) \middle| S_t = s, A_t = a; \mu \right]. \tag{13}$$

   Taking $n \to \infty$, as justified for the corresponding limiting step in the proof of Property 6, gives

$$\begin{aligned}
q^\pi(s,a) &\geq \mathbf{E} \left[\sum_{k=0}^{\infty} \gamma^k R_{t+k} \middle| S_t = s, A_t = a; \mu \right] && (14)\\
&= q^\mu(s,a). && (15)
\end{aligned}$$

   Since this holds for every policy μ, $q^\pi(s,a) \geq \max_\mu q^\mu(s,a) = q^*(s,a)$. The definition of q^* also gives $q^\pi(s,a) \leq q^*(s,a)$. Hence $q^\pi = q^*$.

4. (6 Points) Give an example of an MDP and policy for which the Dynamic Programming Policy Evaluation algorithm presented in class, initialized with $v_0(s) = 0$ for all $s \in \mathcal{S}$, does not reach v^π after any finite number of iterations, but does converge to v^π in the limit. Specify all components of your MDP and the policy. Prove that the algorithm does not converge to v^π in a finite number of iterations, and also prove that it converges to v^π in the limit as the number of iterations goes to infinity. **Hint:** To make the proofs simple, try to make the MDP have as few states and actions as possible.

Let $\mathcal{S} = \{s\}$, $\mathcal{A} = \{a\}$, $p(s, a, s) = 1$, $R(s, a) = 1$, $d_0(s) = 1$, and $\gamma = \frac{1}{2}$. Let rewards always equal 1, and let $\pi(s, a) = 1$. The value of the state is

$$v^\pi(s) = \mathbf{E}\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k} \middle| S_t = s; \pi\right] \tag{16}$$

$$= \sum_{k=0}^{\infty} \gamma^k 1 \tag{17}$$

$$= \sum_{k=0}^{\infty} \left(\frac{1}{2}\right)^k \tag{18}$$

$$= \frac{1}{1 - \frac{1}{2}} \tag{19}$$

$$= 2. \tag{20}$$

Starting with $v_0(s) = 0$, the update is $v_{n+1}(s) = 1 + \frac{1}{2} v_n(s)$. The first few estimates are

$$v_0(s) = 0, \tag{21}$$

$$v_1(s) = 1 + \frac{1}{2} v_0(s) = 1, \tag{22}$$

$$v_2(s) = 1 + \frac{1}{2} v_1(s) = 1 + \frac{1}{2}(1), \tag{23}$$

$$v_3(s) = 1 + \frac{1}{2} v_2(s) = 1 + \frac{1}{2}\left(1 + \frac{1}{2}\right) = 1 + \frac{1}{2} + \frac{1}{4}, \tag{24}$$

$$v_4(s) = 1 + \frac{1}{2} v_3(s) = 1 + \frac{1}{2}\left(1 + \frac{1}{2} + \frac{1}{4}\right) = 1 + \frac{1}{2} + \frac{1}{4} + \frac{1}{8}. \tag{25}$$

The terms in parentheses are the previous estimates. In general, $v_n(s)$ contains the first n terms of the geometric series:

$$v_n(s) = \sum_{k=0}^{n-1} \left(\frac{1}{2}\right)^k. \tag{26}$$

For any finite n, this will be less than 2—it is a finite geometric series and is equal to $2(1 - 2^{-n})$ [link]. This establishes that a finite number of iterations is not enough for the algorithm to converge. Next, to see that it does converge to v^π, notice that the infinite geometric series is equal to 2:

$$\sum_{k=0}^{\infty} \left(\frac{1}{2}\right)^k = 2. \tag{27}$$

Or, to be more precise about convergence,

$$\lim_{n \to \infty} v_n(s) = 2. \tag{28}$$

5. (6 Points) Once you've completed Question 1, look up the Bellman equation for q^π (e.g., ask an AI tool, or check in Sutton and Barto's book). Ensuring that you have the correct form of the Bellman equation for q^π, use it to create a dynamic programming policy evaluation update for estimating q^π. Use q_i to denote the current estimate and q_{i+1} to denote the next estimate. You do not need to prove convergence.

$$q_{i+1}(s,a) = R(s,a) + \gamma \sum_{s' \in \mathcal{S}} p(s,a,s') \sum_{a' \in \mathcal{A}} \pi(s',a') q_i(s',a'). \tag{29}$$

6. (18 Points) Prove the following stochastic policy improvement theorem. Let π and π' be policies, and suppose that for every $s \in \mathcal{S}$,

$$\sum_{a \in \mathcal{A}} \pi'(s,a) q^\pi(s,a) \geq v^\pi(s). \tag{30}$$

Prove that $v^{\pi'}(s) \geq v^\pi(s)$ for every $s \in \mathcal{S}$. Your proof must allow π' to be stochastic.

We follow the unrolling approach from the proof of Property 6 in the course notes. For every state s,

$$\begin{aligned}
v^\pi(s) &\overset{(a)}{\leq} \sum_{a \in \mathcal{A}} \pi'(s,a) q^\pi(s,a) && (31)\\
&\overset{(b)}{=} \sum_{a \in \mathcal{A}} \pi'(s,a) \sum_{s' \in \mathcal{S}} p(s,a,s') \Big(R(s,a) + \gamma v^\pi(s') \Big) && (32)\\
&\overset{(c)}{\leq} \sum_{a \in \mathcal{A}} \pi'(s,a) \sum_{s' \in \mathcal{S}} p(s,a,s') \Big(R(s,a) + \gamma \sum_{a' \in \mathcal{A}} \pi'(s',a') q^\pi(s',a') \Big) && (33)\\
&\overset{(d)}{=} \sum_{a \in \mathcal{A}} \pi'(s,a) \sum_{s' \in \mathcal{S}} p(s,a,s') \Big(R(s,a) && (34)\\
&\quad + \gamma \sum_{a' \in \mathcal{A}} \pi'(s',a') \sum_{s'' \in \mathcal{S}} p(s',a',s'') \Big(R(s',a') + \gamma v^\pi(s'') \Big) \Big). && (35)
\end{aligned}$$

Here **(a)** follows from the assumption of the theorem, **(b)** follows from the relation between q^π and v^π, **(c)** follows by applying the assumption at each next state s', and **(d)** follows by replacing $q^\pi(s',a')$ with its expression in terms of R, p, and v^π.

If we continue this process of expanding the expression using the assumption and the Bellman equation, then after n expansions we have that

$$v^\pi(s) \leq \mathbf{E} \left[\sum_{k=0}^{n-1} \gamma^k R_{t+k} + \gamma^n v^\pi(S_{t+n}) \middle| S_t = s; \pi' \right]. \tag{36}$$

Taking $n \to \infty$ gives

$$\begin{aligned}
v^\pi(s) &\leq \mathbf{E} \left[\sum_{k=0}^{\infty} \gamma^k R_{t+k} \middle| S_t = s; \pi' \right] && (37)\\
&= v^{\pi'}(s). && (38)
\end{aligned}$$

The limiting step is justified in the paragraph following the n-expansion expression in the proof of Property 6 in the course notes. Since this holds for every state s, $v^{\pi'}(s) \geq v^\pi(s)$ for every $s \in \mathcal{S}$.

Part Two: Programming (40 Points)

7. (40 Points) Implement Dynamic Programming Policy Evaluation for the state-value function in any programming language. Your program should read an MDP and a policy from two files, evaluate that policy, and write the resulting value function to an output file. Use the following file formats. 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:

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

   Policy file: The file has n lines and m numbers per line. Line $s+1$ contains $\pi(s,0), \ldots, \pi(s,m-1)$. These are the action probabilities themselves, not parameters to transform using a softmax.

   Output file: Write n lines, with one number per line: $\widehat{v}(0), \ldots, \widehat{v}(n-1)$, in that order. Here $\widehat{v}$ is the estimate of v^π computed using DP policy evaluation. Write at least eight digits after the decimal point.

   Use double precision, initialize every state value to zero, and use the synchronous DP policy evaluation 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.

   First, test your program using `687_gridworld_MDP.txt` and `687_gridworld_policy.txt`, which describe 687-Gridworld and the uniform random policy. The file `687_Gridworld_values.txt` contains the correct output. **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. 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` and `test_policy.txt`. This MDP has seven states and ten actions; every state can transition to every state with positive probability under every action. The supplied policy is stochastic and is not uniform. In your PDF answer, report your estimates of $v^\pi(0), \ldots, v^\pi(6)$, giving at least six digits after the decimal point. You do not need to submit your source code or output files, or report the gridworld values.

s	$\widehat{v}(s)$
0	−0.055232801986
1	−0.004842551037
2	−0.288570522433
3	−2.197646837374
4	−1.134592690258
5	−0.528581180375
6	−0.181693369140