CS7180: Special Topic - Artificial Intelligence (Spring 2022) Christopher Amato
Northeastern University Due Feb 9, 2022

Exercise 2: Markov Decision Processes (MDPs)

Please remember:

- Exercise due at **11:59 PM EST Feb 9, 2022**.
- Submissions should be made electronically on Canvas. Please ensure that your solutions for both the written and programming parts are present. You can upload multiple files in a single submission, or you can zip them into a single file. You can make as many submissions as you wish, but only the latest one will be considered.
- For **Written** questions, solutions may be handwritten or typeset. If you write your answers by hand and submit images/scans of them, please please ensure legibility and order them correctly in a single PDF file.
- The PDF file should also include the figures from the **Plot** questions.
- For both **Plot** and **Code** questions, submit your source code along with reasonable documentation.
- You are welcome to discuss these problems with other students in the class, but you must understand and write up the solution and code yourself. Also, you *must* list the names of all those (if any) with whom you discussed your answers at the top of your PDF solutions page.
- Each exercise may be handed in up to two days late (24-hour period), penalized by 10% per day late. Submissions later than two days will not be accepted.
- Contact the teaching staff if there are medical or other extenuating circumstances that we should be aware of.

1. **1 point.** *Formulating an MDP.*
   Written: It is instructive to formally define an MDP at least once, in particular, the dynamics function. Consider the four-rooms domain from Ex0.
   (a) What are the state and action spaces S, A?
   (b) Consider the dynamics function $p(s', r|s, a)$. Approximately how many non-zero rows are in this conditional probability table?

2. **1 point.** (RL2e 3.6, 3.7) *The RL objective.*
   Written:
   (a) Suppose you treated pole-balancing as an episodic task but also used discounting, with all rewards zero except for -1 upon failure. What then would the return be at each time? How does this return differ from that in the discounted, continuing formulation of this task? (Derive expressions for both the episodic and continuing cases.)
   (b) Imagine that you are designing a robot to run a maze. You decide to give it a reward of $+1$ for escaping from the maze and a reward of zero at all other times. The task seems to break down naturally into episodes – the successive runs through the maze – so you decide to treat it as an episodic task, where the goal is to maximize expected total reward (Equation 3.7). There is no discounting, i.e. $\gamma = 1$. After running the learning agent for a while, you find that it is showing no improvement in escaping from the maze. What is going wrong? Have you effectively communicated to the agent what you want it to achieve?

3. **1 point.** (RL2e 3.15, 3.16) *Modifying the reward function.*
   Written:
   (a) In the gridworld example (Figure 3.2 in RL2e), rewards are positive for goals, negative for running into the edge of the world, and zero the rest of the time. Are the signs of these rewards important, or only the intervals between them? Prove, using Equation 3.8, that adding a constant c to all the rewards adds a constant, v_c, to the values of all states, and thus does not affect the relative values of any states under any policies. What is v_c in terms of c and γ?
   (b) Now consider adding a constant c to all the rewards in an episodic task, such as maze running. Would this have any effect, or would it leave the task unchanged as in the continuing task above? Why or why not? Give an example.

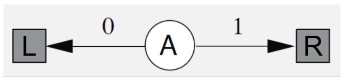
4. **2 point.** (RL2e 3.14) *Bellman equation.*
Written:

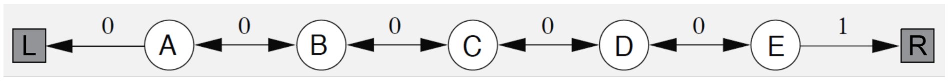
   (a) The Bellman equation (Equation 3.14) must hold for each state for the value function v_π shown in Figure 3.2 (right) of Example 3.5. Show numerically that this equation holds for the center state, valued at $+0.7$, with respect to its four neighboring states, valued at $+2.3, +0.4, -0.4, +0.7$. (These numbers are accurate only to one decimal place.) Note that Figure 3.2 (right) is the value function for the equiprobable random policy.

   (b) The Bellman equation holds for *all* policies, including optimal policies. Consider v_* and π_* shown in Figure 3.5 (middle, right respectively). Similar to the previous part, show numerically that the Bellman equation holds for the center state, valued at $+17.8$, with respect to its four neighboring states, for the optimal policy π_* shown in Figure 3.5 (right).

If the Bellman equation is unclear to you, you should practice this question again for other states.

5. **1 point.** *Guessing and verifying value functions.*





Written: We have not discussed algorithms for finding the value function yet, but it is possible to guess them for simple problems and verify that they are consistent with the Bellman equation. A purported value function $V(s)$ is consistent (according to the Bellman equation) for all states under the policy π *if and only if* $V = v_\pi$, the unique value function for π.

   (a) Consider the simple 3-state MDP shown above (left). All episodes start in the center state, A, then proceed either left or right by one state on each step, with equal probability. Episodes terminate either on the extreme left (L) or the extreme right (R). When an episode terminates on the right, a reward of $+1$ occurs; all other rewards are zero. For instance, taking the action right in state A transitions to R, provides a reward of 1 and terminates the episode. This is an undiscounted MDP ($\gamma = 1$). Guess the value function for this MDP (for the equiprobable random policy), and verify its consistency using the Bellman equation.

   (b) Now consider an extension of the MDP to contain 7 states total (right). Guess the value function for this MDP (for the equiprobable random policy), and verify its consistency using the Bellman equation.
   Hint: The new value function has a simple form. Use part (a) to inform your thinking.

   (c) What do you think the value function is for arbitrary an arbitrary number of states n?

6. **2 points.** *Solving for the value function.*
Written: Another way to solve for the value function is to write out the Bellman equation for each state, view it as a system of linear equations, and then solve for the unknowns (the value of each state). Consider a particularly simple MDP, the 2-state recycling robot in Example 3.3.

   (a) Expand the Bellman equation for the 2 states in the recycling robot, for an arbitrary policy $\pi(a|s)$, discount factor γ, and domain parameters $\alpha, \beta, r_{\texttt{search}}, r_{\texttt{wait}}$ as described in the example.

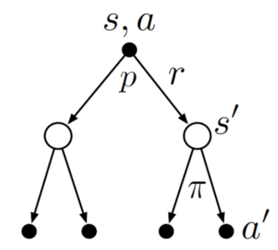
   (b) You should now have two linear equations involving two unknowns, $v(\texttt{high})$ and $v(\texttt{low})$, as well as involving the policy $\pi(a|s)$, γ, and the domain parameters. Let $\alpha = 0.8, \beta = 0.6, \gamma = 0.9, r_{\texttt{search}} = 10, r_{\texttt{wait}} = 3$.
   Consider the policy $\pi(\texttt{search} \,|\, \texttt{high}) = 1$, $\pi(\texttt{wait} \,|\, \texttt{low}) = 0.5$, and $\pi(\texttt{recharge} \,|\, \texttt{low}) = 0.5$.
   Find the value function for this policy, i.e., solve the equations for the values of $v(\texttt{high})$ and $v(\texttt{low})$.
   Check that your solution satisfies the Bellman equation.

   (c) (**Bonus, optional**) Suppose you can modify the policy in the `low` state, i.e., set $\pi(\texttt{wait} \,|\, \texttt{low}) = \theta$, and $\pi(\texttt{recharge} \,|\, \texttt{low}) = 1 - \theta$. What θ should you set it to, and what is the value function for that θ?

7. **2 point.** (RL2e 3.12, 3.13, 3.17) *Action-value function.*

Written:

   (a) Give an equation for v_π in terms of q_π and π.



q_π backup diagram

(b) Give an equation for q_π in terms of v_π and the four-argument p.

(c) What is the Bellman equation for action values, that is, for q_π? It must give the action value $q_\pi(s, a)$ in terms of the action values, $q_\pi(s', a')$, of possible successors to the state–action pair (s, a).
Hint: The backup diagram above corresponds to this equation. Show the sequence of equations analogous to Equation 3.14, but for action values.