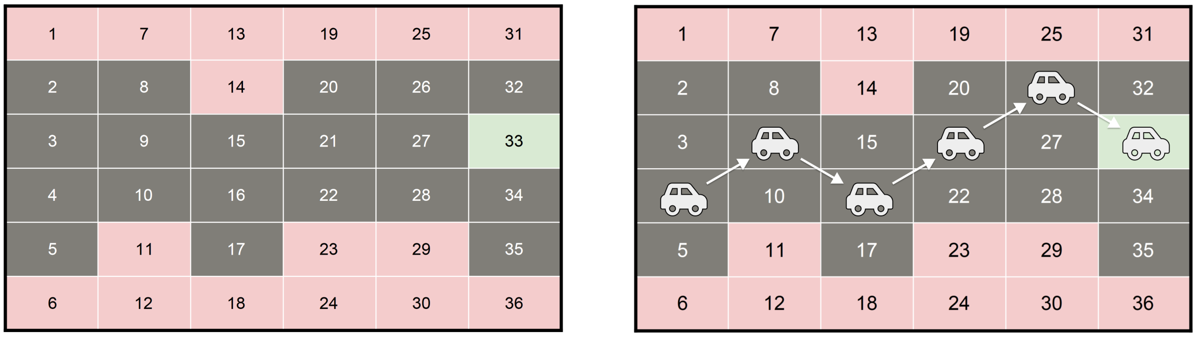
CS 234 Winter 2022
Assignment 1
Due: January 14 at 6:00 pm (PST)

For submission instructions, please refer to website. For all problems, if you use an existing result from either the literature or a textbook to solve the exercise, you need to cite the source.

1 Efficient Routing MDP [35 pts]

You are leading a routing and planning team at a self-driving car company and have decided to model your latest urban navigation problem as an MDP. Consider the following environment (Fig. 1). Your car must navigate along the road (gray squares) while avoiding obstacles (red squares) to reach the rider's destination (the green square). Because the road is gridlocked, your car must change lanes whenever it wishes to move forward. From any gray square, your car can either move right & up, or right & down. For example, starting from state 3, your car can move to state 8 or 10. Note that it is not be possible to reach the green square from every state. Actions are deterministic and always succeed unless they will cause you to run into an impassible barrier. The thick outer edge indicates an impassible barrier, and attempting to move in the direction of a barrier from a gray square results in your car moving up one square (e.g. taking any action from state 32 moves the car to state 31).



(a) Grid World

(b) A successful run in Grid World.

Figure 1

A successful run in Grid World 1 is shown in Figure 1b. Taking any action from the green destination square (no. 33) earns a reward of r_g and ends the episode. Taking any action from the red squares that depict obstacles (no. 1, 7, 13...) earns a reward of r_r and ends the episode.

Otherwise, from every other square, taking any action is associated with a reward r_s. Assume the discount factor $\gamma = 0.9$, $r_g = +5$, and $r_r = -5$ unless otherwise specified. Notice the horizon is technically infinite.

(a) Let $r_s \in \{-5, -0.5, 0, 2\}$. Starting in **square 2**, for each of the possible values of r_s, briefly explain what the optimal policy would be in Grid World. In each case is the optimal policy unique and does the optimal policy depend on the value of the discount factor γ? Explain your answer. [5 pts]

First, note that under the inefficient routing, there are three possible paths to the green 33, namely

$$P_{s,1} = [2, 9, 16, 21, 26, 33],$$

$$P_{s,2} = [2, 9, 16, 21, 28, 33]$$

and

$$P_{s,3} = [2, 9, 16, 21, 28, 35, 34, 33].$$

We see that $P_{s,1}$ and $P_{s,2}$ are the shortest such paths (using up 5 grey steps), while $P_{s,3}$ is the longest, using up 7 grey steps. Of note, $P_{s,3}$ technically describes four distinct policies: an attempted right + down or attempted right + up at 35, combined with an attempted right + down or attempted right + up at 34. However, without loss, we'll treat these policies as the same, as however one runs into the barriers at the end doesn't shape the overall findings.

Lastly, note that when $\gamma = 0$, any policy is optimal, as there are no discounted future rewards (all zero'd out), so the player gets whatever points for moving off of 2 and that's it. So in this way, things clearly depend on γ, though admittedly this is a bit of a trivial case.

Case I: $r_s = -5$ In this case, reward is maximized by minimizing penalty, namely escaping the grid as fast as possible by moving $2 \to 7 \to END$, for a total of $-5(1+\gamma) = -9.5$ points. For any $\gamma > 0$, cutting losses and escaping will be the optimal policy, so optimality here does not depend on γ (apart from the trivial case). Further, the path is unique, as it is the single fastest way out of the game.

Case II: $r_s = -.5$ In this case, two possibilities present themselves: (i) get out of the game immediately to stop the accrual of negative discounted rewards, or (ii) take some pain (using one of the shortest paths ($P_{s,1}$ and $P_{s,2}$) on the way to 33, in the hopes that the discounted reward of 33 will offset all the (discounted) pain to get there. In the case of (i), the reward is $-.5 - \gamma \cdot 5$, while in the case of (ii), the reward is $\left(-.5 \sum_{i=0}^{4} \gamma^i \right) + \gamma^5 \cdot 5$. This allows us to solve for the breakeven, i.e.

$$-.5 + -5\gamma < \underbrace{\left(-.5 \sum_{i=0}^{4} \gamma^i \right)}_{\text{Disc. Rew. of } P_{s,1} \text{ or } P_{s,2}} + \gamma^5 \cdot 5.$$

This inequality is always satisfied for $\gamma \in (0, 1]$, so it always makes sense to leg it out for a total of .9049 and follow either of $P_{s,1}$ or $P_{s,2}$. Those are the equally-optimal policies, so no uniqueness. Further, as the breakeven equation shows, the optimal policy does not depend on γ (as there is no breakeven point in $(0, 1]$).

Case III: $r_s = 0$ There are two cases to consider here, in addition to the trivial $\gamma = 0$ case presented at the outset. When $\gamma < 1$, we want to navigate to the green 33 as fast as possible, lest the final $5 \cdot \gamma^j$ be discounted any more than it need be. In this case, we again follow one of $P_{s,1}$ or $P_{s,2}$ (the shortest/fastest paths to green from 2) to achieve a reward of $0 \cdot \sum_{i=0}^{4} \gamma^i + 5\gamma^5 = 2.95245$. So optimality is not unique for $\gamma = 0.9$.

Notably, when $\gamma = 1$, we can take as long as we want to reach 33, as 33 will not be discounted at the end. Thus, in this case, $P_{s,3}$ suffices as well. So we see that optimality is not only not unique, but it also depends on γ.

Case IV: $r_s = 2$ There are two routes to consider here: (i) go straight to 33, accruing fewer grey 2's but a less discounted 5, or (ii) take as long as possible to get to 33, accruing maximum grey 2's but a more discounted 5. In the first case, we have

$$G(P_{s,1}) = P_{s,2} = \left(2 \sum_{i=0}^{4} \gamma^i\right) + \gamma^5 \cdot 5,$$

while

$$G(P_{s,1}) = P_{s,2} = \left(2 \sum_{i=0}^{6} \gamma^i\right) + \gamma^7 \cdot 5.$$

Leaving γ still arbitrary, this produces a breakeven point of

$$\left(2 \sum_{i=0}^{6} \gamma_*^i\right) + \gamma_*^7 \cdot 5 = \left(2 \sum_{i=0}^{4} \gamma_*^i\right) + \gamma_*^5 \cdot 5 \implies \gamma_* = 3/5,$$

showing that optimality does in fact depend on γ. So for $\gamma = 0.9 > 3/5$ here, we want to go the long route (total value 12.8255). That is, $P_{s,3}$ is the optimal route here. In terms of uniqueness, it is unique in the points that it visits, though there is room for different actions (as set forth at the very start of this problem) at points 35 and 34. So the specific policy is not unique (as actions differ on the bounds), but the overall path is.

(b) Which values of $r_s \in \{-5, -0.5, 0, 2\}$ will yield a policy that returns the shortest path to the green square? (Hint: At least one does.) Explain which ones do, then, pick the minimum of this set of rewards that does, and then find the optimal value function for **states 2, 13, 21 and 32**. [5 pts]

As set forth above, the -1/2 and 0 cases both encourage one of the shortest paths; however, the -1/2 case returns a reward of .9049, which is less than the reward for 0 which is 2.95245. So we'll consider the $r_s = -1/2$ case here. Using our deterministic/optimal policy, we already know that $V(2) = .9049 here$. Then, we have

$$\begin{aligned} V(21) &= R(21) + \gamma V(28) \\ &= R(21) + \gamma R(28) + \gamma^2 V(33) \\ &= R(21) + \gamma R(28) + \gamma^2 R(33) \\ &= -0.5 - (0.9) \cdot (.5) + (0.81) \cdot 5 \\ &= 3.1. \end{aligned}$$

We did not cover the optimal policy for 32 above, but there is only one move to be made, and that is up (at which point game over). Thus,

$$\begin{aligned} V(32) &= R(32) + \gamma V(31) \\ &= R(32) + \gamma R(31) \\ &= -.5 - .9 \cdot (5) \\ &= -5.0. \end{aligned}$$

We also did not cover 13, but it's game over from the start, and hence $V(13) = R(13) = -5.0$

Finally, from 2, we have using our result above

$$\begin{aligned} V(2) &= R(2) + \gamma V(9) \\ &= R(2) + \gamma R(9) + \gamma^2 V(16) \\ &= R(2) + \gamma R(9) + \gamma^2 R(16) + \gamma^3 V(21) \\ &= -0.5 - 0.9 \cdot (.5) - 0.81 \cdot (.5) + 0.729 \cdot (3.1) \\ &= 0.9049. \end{aligned}$$

Rather than finding the shortest path between two points, suppose our car is low on gas, so we want to take the path that uses the least fuel. In the real world, navigation optimized for fuel consumption may take more steps to reach a destination [1].

Consider the same MDP, but with two new "efficient actions" – move right or move down. For example, starting from state 3, you can either move to state 4 or 9. Once again, the actions are deterministic and always succeed unless you run into a wall. Attempting to move in the direction of a wall from a gray square using an efficient action results in you moving *down* one square. For clarity, we will use separate symbols r_s for the reward associated with an inefficient action (right & up, or right & down) and r_e for the reward associated with an efficient action.

(c) Let $r_e \in \{-5, -0.5, 0, 2\}$. Starting in **state 2**, for each of the possible values of r_e, briefly explain what the optimal policy would be in Grid World *using only efficient actions*. In each case is the optimal policy unique and does the optimal policy depend on the value of the discount factor γ? Explain your answer. Which values of r_e would cause the optimal policy to return the shortest path to the green destination square? [5 pts]

[1] Google Maps Blog

Now, there are two viable paths from 2 to 33. First, there is

$$P_{e,1} = [2, 3, 9, 15, 21, 27, 33]$$

and

$$P_{e,2} = [2, 8, 9, 15, 21, 27, 33].$$

Following either one of these paths nets a discounted sum of rewards of

$$r_e \sum_{i=0}^{5} +5 \cdot \gamma^6,$$

as both make six grey moves before a final green move. Critically, these are the *only* two paths to 33; once you go below, you can never recover upwards.

Case I: $r_e = -5$
This one comes down to cutting losses and exiting directly right (2 moves to red 14), or going to the green square (6 total – 1 down, 5 right – moves to reach green). The breakeven equation is

$$-5 \sum_{i=0}^{2} \gamma_*^i = -5 \sum_{i=0}^{5} \gamma_*^i + 5\gamma_*^6,$$

which gives a breakeven point outside of (0, 1]. Thus, optimal path does not depend on γ, as it always favors to exit directly right, through 14. Clearly, this is a unique path. Value is $-5 - .9 * 5 - .81 * 5 = -13.55$

Case II: $r_e = -.5$
Again, it's a choice between: (i) directly right/out through 14 or (ii) reach-the-green via $P_{e,1}$ or $P_{e,2}$. The breakeven equation is then

$$-.5 \sum_{i=0}^{1} \gamma_*^i - 5\gamma_*^2 = \left(-.5 \sum_{i=0}^{5} \gamma_*^i \right) + 5\gamma_*^6,$$

which gives no breakeven point in [0, 1]. This means that optimal policy does not depend on γ, and it is preferable to move to the green. Accordingly, optimal policy is necessarily one of of $P_{e,1}$ or $P_{e,2}$ (so not unique), with value $\left(-.5 \sum_{i=0}^{5} (.9)^i \right) + 5(.9)^6 = .31441$

Case III: $r_e = 0$
Here, discounted reward is maximized by reaching the green, to guarantee a positive value. Since the only two possible paths to the green ($P_{e,1}$ or $P_{e,2}$) are of equal distance, their reward is the same and through lack of competition, they are the optimal paths. Moreover, there as no dependence on γ, since the only viable paths are of equal distance and thus accrue identical discounts. So the optimal path is one of $P_{e,1}$ or $P_{e,2}$; optimality doesn't depend on γ and optimal paths are not unique.

Case IV: $r_e = 2$
Again, discounted reward is maximized by reaching the green, to guarantee a positive value.

Since the only two possible paths to the green ($P_{e,1}$ or $P_{e,2}$) are of equal distance, their reward is the same and through lack of competition, they are the optimal paths. Moreover, there as no dependence on γ, since the only viable paths are of equal distance and thus accrue identical discounts. As before, the optimal path is one of $P_{e,1}$ or $P_{e,2}$; optimality doesn't depend on γ and optimal paths are not unique.

(d) Consider now that $r_s = 0$. Derive a relation for r_e such that the optimal path from **state 2** to the destination square using only efficient actions is strictly more rewarding than the optimal path using only inefficient actions. [5 pts]

As set forth above, the inefficient reward for $\gamma = .9$ and $r_s = 0$ is $5 \cdot (.9)^5 = 2.95245$. Hence, we need r_e such that the reward for $P_{e,1}$ and (identically) $P_{e,2}$ is greater than this 2.95245. Since there is only efficient one path to green from 2 to consider, we just solve

$$2.95245 < \left(r_e \sum_{i=0}^{5} (.9)^i \right) + 5(.9)^6$$

$$\implies$$

$$r_e > \frac{5 \cdot (.9)^5 - 5 \cdot (.9)^6}{\sum_{i=0}^{5} (.9)^i} \approx 0.0630113.$$

(e) Compare the set of gray states that can reach the goal using only efficient actions, and the set of gray states that can reach the goal using only inefficient actions. Which states are part of one set but not part of the other? Explain your answer. [5 pts]

Using right-diagonal, i.e. inefficient, actions, we first see that the penultimate states 26, 27, 28, and 34[2] all directly lead to the target state; accordingly, 35[3], as well as 20, 21, 22, can reach these penultimate states and thus the goal. Working leftwards, all of 15-17, and then 8-10, and then 2-5 can then reach the third-to-last states. Accordingly, all grey states **except 32** can reach the goal. 32 fails because the only inefficient move is out of bounds, and then up into a red state.

In contrast, for efficient actions, any of 2, 8, 20, 26, and 32 can reach the target state by a downward move, followed by zero or more rightward moves. Further, any of 3, 9, 15, 21, and 27 can reach the target state just by rightward actions. And sadly, each of 4, 10, 16, 22, 28, 34, 5, 17, and 35 **cannot** reach the target state, as there is no upward move form them to recover to the third row down: if they go out of bounds, they go down again. Hence, we have

(a) **Reachable Under Inefficient but not Efficient:** 4, 10, 16, 22, 28, 34, 5, 17, 35.

(b) **Reachable Under Efficient but not Inefficient:** 32.

(c) **Reachable Under Both**: 2, 8, 20, 26, 3, 9, 15, 21, 27.

(f) Consider a general MDP with rewards and transitions. Consider a discount factor of γ. Assume that the horizon is infinite (so there is no termination). Can adding a constant c to all rewards

[2] Go out of bounds, then up

[3] Again go out and up

($r_{new} = c + r_{old}$) change the optimal policy of the MDP? If yes, give an example for Grid World with efficient actions using the r_g, r_s and r_e such that the optimal policy changes for a specific constant. [5 pts]

The following setup outlines a scenario for which the addition of c changes optimality. Consider the efficient grid world (as specified above) in which $r_e = -5$ and $\gamma = 1$, and you are starting from state 2. As set forth in part c, the optimal path is to exit the game as soon as possible, by moving rightwards through 14. However, suppose we add 5 to every reward, making the "new" rewards $r'_s = r'_r = 0$ and $r'_g = 10$. Now, rewards are no longer a form of penalty, and there is no incentive to leave the game early. Whereas the original rightwards (exit ASAP) path would now get you $5 + 1 \cdot 5 + 1 \cdot 5$, either of $P_{e,1}$ or $P_{e,2}$ would score you a whopping 10 points under the new construction. Hence, the optimal path would shift from the $[2, 8, 14]$ to the familiar $P_{e,1}$ or $P_{e,2}$, showing that the inclusion of a constant could in fact change the landscape here for this specific constant.

(g) Imagine your efficient routing MDP is to be used in a popular maps app or website. Choosing what route options will be available and which will be default present inevitable value judgements. The shortest path and the sustainable path are each optimal given their choices of rewards, but the rewards formalize different values. How would you present the shortest path and sustainable path options in a maps app and why? Please use 2-4 sentences to explain your answer. There is no single "correct" answer– reasonable explanations of your choice will receive full credit. [5 pts]

As this is a maps app to be used by the public, it would be imperative to keep my presentation of the value judgments as high-level and non-technical as possible, so that those without any statistics/probability/RL background could use the app on equal footing. To this end, the "Goals" on slides 41-44 of Lecture 1 provide a digestible and accessible framework for conveying the context of optimality here. Specifically (plagiarizing slides 41-44), I might say that the shortest path option selects actions that minimize total (expected) driving time (i.e. maximizes time *not* in traffic; assume generally that each grey block requires a unit of time to traverse) to the destination, secondary/tie-breaking consideration for fuel efficiency. By contrast, I might say that the sustainable path option minimizes fuel/energy consumption (i.e. maximizes fuel efficiency), secondary/tie-breaking consideration for time. In other words, the app will narrow down your paths by what's fastest or most fuel-saving; from there, it'll (respectively) choose candidates based on what is most efficient or faster.

Context for the debate: "Green nudges" such as setting the most efficient route as the (changeable) default on mapping apps have been proposed as a way to encourage environmentally beneficial actions.[4] They can help close the gap between the desire to act more sustainably that many people express and their day to day behavior.[5] Some have argued that green nudges do not go far enough and that given the urgency of climate change the sustainable option should be the only one presented. Others argue that nudges infringe autonomy and so users should be explicitly asked which default they would prefer; others that nudges are only acceptable when the intention behind the nudge is transparently presented. For more context, see [5].

[4] A nudge is an "aspect of choice architecture that alters people's behaviour in a predictable way without forbidding any options or significantly changing economic incentives" (Thaler and Sunstein, 2008).

[5] Siipi and Koi, 2021. https://philpapers.org/archive/SIITEO.pdf

2 Value Iteration Theorem [35 pts]

In this problem, we will deal with contractions and fixed points and prove an important result from the value iteration theorem. From lecture, we know that the Bellman backup operator B given below is a contraction with the fixed point as V^*, the optimal value function of the MDP. The symbols have their usual meanings. γ is the discount factor and $0 \leq \gamma < 1$. In all parts, $||v||$ is the infinity norm of the vector.

$$(BV)(s) = \max_a [R(s,a) + \gamma \sum_{s' \in S} p(s'|s,a)V(s')]$$

We also saw the contraction operator B_π which is the Bellman backup operator for a particular policy given below:

$$(B_\pi V)(s) = \mathbb{E}_{a \sim \pi}[R(s,a) + \gamma \sum_{s' \in S} p(s'|s,a)V(s')]$$

(a) Recall that $||BV - BV'|| \leq \gamma ||V - V'||$ for two random value functions V and V'. Prove that B_π is also a contraction mapping: $||B_\pi V - B_\pi V'|| \leq \gamma ||V - V'||$. [5 pts] This proof is quite similar to the contraction proof given in class. Specifically, we do:

$$\begin{aligned}
||B_\pi V(s) - B_\pi V'(s)|| &= \left\| \mathbb{E}_{a \sim \pi}[R(s,a) + \gamma \sum_{s' \in S} p(s'|s,a)V(s')] - \mathbb{E}_{a \sim \pi}[R(s,a) + \gamma \sum_{s' \in S} p(s'|s,a)V'(s')] \right\| \\
&= \left\| \mathbb{E}_{a \sim \pi}[R(s,a) + \gamma \sum_{s' \in S} p(s'|s,a)V(s') - R(s,a) - \gamma \sum_{s' \in S} p(s'|s,a)V'(s')] \right\| \\
&= \left\| \mathbb{E}_{a \sim \pi}[\gamma \sum_{s' \in S} p(s'|s,a)(V(s') - V'(s'))] \right\| \\
&\leq \left\| \mathbb{E}_{a \sim \pi}[\gamma \sum_{s' \in S} p(s'|s,a)||V(s') - V'(s')||] \right\| \\
&= \gamma ||V(s') - V'(s')|| \left\| \mathbb{E}_{a \sim \pi}[\sum_{s' \in S} p(s'|s,a)] \right\| \\
&= \gamma ||V(s') - V'(s')|| \left\| \underbrace{\sum_{a \in A} \sum_{s' \in S} \pi(a|s)(p(s'|s,a)}_{1} \right\| \\
&= \gamma ||V(s') - V'(s')|| ||1|| \\
&= \gamma ||V(s') - V'(s')||,
\end{aligned}$$

as desired.

(b) Prove that the fixed point for B_π is unique. What is the fixed point of B_π? [5 pts] The fixed point for B_π is V_π, which we know satisfies the Bellman for any state s:

$$V_\pi(s) = B_\pi V_\pi(s) = \mathbb{E}_{a \sim \pi}[R(s,a) + \gamma \sum_{s' \in S} p(s'|s,a)V_\pi(s').$$

To see uniqueness, suppose, for the purposes of a contradiction, that there are two *unequal* fixed points $V_\pi \neq V'_\pi$. Accordingly, we will have, by virtue of both being fixed points and the contraction finding above:

$$\begin{aligned}||V_\pi - V'_\pi|| &= ||B_\pi V_\pi - B_\pi V'_\pi|| \\ &\leq \gamma ||V_\pi - V'_\pi||.\end{aligned}$$

Now, we know that $\gamma \in [0, 1)$, so there are two scenarios. First, if $\gamma = 0$, then

$$||V_\pi - V'_\pi|| = 0 \implies V_\pi = V'_\pi,$$

which is a contradiction (i.e. the two must actually equal). Second, if $\gamma \in (0, 1)$, then

$$||V_\pi - V'_\pi|| \leq \gamma ||V_\pi - V'_\pi|| \implies V_\pi = V'_\pi,$$

which is similarly a contradiction. Hence, it cannot be the case that there are two fixed points, so the fixed point is unique.

In value iteration, we repeatedly apply the Bellman backup operator B to improve our value function. At the end of value iteration, we can recover a greedy policy π from the value function using the equation below:

$$\pi(s) = \arg\max_a [r(s, a) + \gamma \sum_{s' \in S} p(s'|s, a)V(s')]$$

Suppose we run value iteration for a finite number of steps to obtain a value function V (V has not necessarily converged to V^*). Say now that we evaluate our policy π obtained using the formula above to get V^π. **Note that here and for the rest of Q2, π refers to the greedy policy.**

(c) Is V_π always the same as V? Justify your answer. [5 pts]

Consider the following counter example. Consider a world with two states, s_1 and s_2 and two actions a_1, a_2 such that $R(s_1, a_1) = R(s_1, a_2) = 2$ and $R(s_2, a_1) = R(s_2, a_2) = 4$ and

$$P_{a_1} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

as well as

$$P_{a_2} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

In short, it's a two-state world, with two sets of actions: the first action takes you to the other state, the second action keeps you at your current state. Now let's do a single round of value iteration, supposing arbitrarily that $\gamma = 1/2$. As prescribed by value iteration, we init with

$$V_0 = [0, 0].$$

A single round of value iteration gives us

$$\begin{aligned}V_1(s_1) &= \max\{R(s_1, a_1) + .5 \cdot V_0(2), R(s_1, a_2) + .5 \cdot V_0(1)\} \\ &= \max\{2 + 0, 2 + 0)\} \\ &= 2,\end{aligned}$$

as well as

$$\begin{aligned}V_1(s_2) &= \max\{R(s_2,a_1) + .5 \cdot V_0(1), R(s_2,a_2) + .5 \cdot V_0(2)\}\\ &= \max\{4+0, 4+0\}\\ &= 4.\end{aligned}$$

At this point, we have $R = [2,4]$ and $V_1 = [2,4]$, so clearly, π here will be $\pi(s_1) : s_1 \to s_2$ and $\pi(s_2) : s_2 \to s_2$. That is, we go to the second state, which has higher reward and a higher value function. Now, when we compute V_π, we will have $V_\pi = [2 + .5 \cdot 4, 4 + .5 \cdot 4] = [4,6]$, and thus we see a case where

$$V_\pi \neq V.$$

In lecture, we learned that running value iteration until a certain tolerance can bring us close to recovering the optimal value function. Let V_n and V_{n+1} be the outputs of value iteration at the n^{th} and $n+1^{th}$ iterations respectively. Let $\varepsilon > 0$ and consider the point in value iteration such that $||V_{n+1} - V_n|| < \frac{\varepsilon(1-\gamma)}{2\gamma}$. Let π be the greedy policy given the value function V_{n+1}.
You will now prove that this policy π is ε-optimal. This result justifies why halting value iteration when the difference between success iterations is sufficiently small, ensures the decision policy obtained by being greedy with respect to the value function, is near-optimal.
Precisely if

$$||V_{n+1} - V_n|| < \frac{\varepsilon(1-\gamma)}{2\gamma}$$

then,

$$||V_\pi - V^*|| \leq \varepsilon$$

(d) When π is the greedy policy, what is the relationship between B and B_π? [2 pts]

Using our definitions above, let's first compare the relationship between B_π for a general policy π with B, before considering the greedy policy. Recall we have

$$(BV)(s) = \max_a [R(s,a) + \gamma \sum_{s' \in S} p(s'|s,a)V(s')].$$

For notation, let $a*$ be the action such that

$$\max_a [R(s,a) + \gamma \sum_{s' \in S} p(s'|s,a)V(s')] = R(s,a^*) + \gamma \sum_{s' \in S} p(s'|s,a^*)V(s'),$$

i.e. a is the value-maximizing action for the state s. By definition, this means that for any action a', we are guaranteed

$$R(s,a') + \gamma \sum_{s' \in S} p(s'|s,a')V(s') \leq R(s,a^*) + \gamma \sum_{s' \in S} p(s'|s,a^*)V(s')$$

In turn, we then get (where $A(\pi)$ is the set of actions you might take under your policy π):

$$\begin{aligned}
(B_\pi V)(s) &= \mathbb{E}_{a\sim\pi}[R(s,a) + \gamma \sum_{s' \in S} p(s'|s,a)V(s')] \\
&= \sum_{a' \in A(\pi)} \pi(a'|s)[R(s,a') + \gamma \sum_{s' \in S} p(s'|s,a')V(s')] \\
&\leq \sum_{a' \in A(\pi)} \pi(a'|s)[R(s,a^*) + \gamma \sum_{s' \in S} p(s'|s,a^*)V(s')] \\
&= [R(s,a^*) + \gamma \sum_{s' \in S} p(s'|s,a^*)V(s')] \sum_{a' \in A(\pi)} \pi(a'|s) \\
&= R(s,a^*) + \gamma \sum_{s' \in S} p(s'|s,a^*)V(s') \\
&= BV(s).
\end{aligned}$$

So in an arbitrary sense, we get this inequality. But let's now think about what π does under the greedy policy. Under the greedy policy, we have that

$$\pi(s) = \arg\max_a [r(s,a) + \gamma \sum_{s' \in S} p(s'|s,a)V(s')]$$

, which means necessarily that greedy π returns a'^* with probability one, (i.e. putting unit density on this a^*, and zero density on all other $A(\pi)$), in order to achieve the requisite maximization. In turn, we have (henceforth, π is greedy)

$$\begin{aligned}
(B_\pi V)(s) &= \mathbb{E}_{a\sim\pi}[R(s,a) + \gamma \sum_{s' \in S} p(s'|s,a)V(s')] \\
&= \underbrace{\pi(a^*|s)}_{1}[R(s,a^*) + \gamma \sum_{s' \in S} p(s'|s,a^*)V(s')] \\
&\quad + \sum_{a' \in A(\pi): a' \neq a^*} \underbrace{\pi(a'|s)}_{0}[R(s,a') + \gamma \sum_{s' \in S} p(s'|s,a')V(s')] \\
&= R(s,a^*) + \gamma \sum_{s' \in S} p(s'|s,a^*)V(s') \\
&\quad + \sum_{a' \in A(\pi): a' \neq a^*} 0[R(s,a') + \gamma \sum_{s' \in S} p(s'|s,a')V(s')] \\
&= R(s,a^*) + \gamma \sum_{s' \in S} p(s'|s,a^*)V(s') \\
&= BV(s).
\end{aligned}$$

So in this greedy case, the two are actually equal.

(e) Prove that $||V_\pi - V_{n+1}|| \leq \varepsilon/2$.
Hint: Introduce an in-between term and leverage the triangle inequality. [6 pts]

We have, using our work up to this point and the premises provided:

$$\begin{aligned}
||V_\pi - V_{n+1}|| &= ||V_\pi - BV_{n+1} + BV_{n+1} - V_{n+1}|| \\
&\leq ||V_\pi - BV_{n+1}|| + ||BV_{n+1} - V_{n+1}|| \\
&= ||V_\pi - BV_{n+1}|| + ||BV_{n+1} - V_{n+1}|| \\
&= ||V_\pi - BV_{n+1}|| + ||BV_{n+1} - BV_n|| \\
&= ||B_\pi V_\pi - BV_{n+1}|| + ||BV_{n+1} - V_{n+1}|| \tag{1} \\
&= ||B_\pi V_\pi - B_\pi V_{n+1}|| + ||BV_{n+1} - V_{n+1}|| \tag{2} \\
&\leq \gamma||V_\pi - V_{n+1}|| + \gamma||V_{n+1} - V_n|| \tag{3} \\
&\leq \gamma||V_\pi - V_{n+1}|| + \gamma\frac{\varepsilon(1-\gamma)}{2\gamma}, \tag{4}
\end{aligned}$$

where the critical steps are: (1), which uses the fixed point finding of B_π in b; (2), which uses the equality in d; (3) which uses the contractions we have found for the Bellman operators; and (4), which is a supposition of the proof. Next, we solve for $||V_\pi - V_{n+1}||$, i.e.

$$||V_\pi - V_{n+1}|| \leq \gamma||V_\pi - V_{n+1}|| + \gamma\frac{\varepsilon(1-\gamma)}{2\gamma}$$

$$\Longrightarrow$$

$$||V_\pi - V_{n+1}||(1-\gamma) \leq \frac{\varepsilon(1-\gamma)}{2}$$

$$\Longrightarrow$$

$$||V_\pi - V_{n+1}|| \leq \frac{\varepsilon}{2},$$

as desired.

(f) Prove $||B^kV - B^kV'|| \leq \gamma^k||V - V'||$ [3 pts] We solve inductively. For a base case k = 1, we have from our in-class results that

$$||B^1V - B^1V'|| \leq \gamma^1||V - V'||.$$

Now, as an inductive hypothesis, suppose it is the case that for arbitrary ℓ,

$$||B^\ell V - B^\ell V'|| \leq \gamma^\ell||V - V'||.$$

We then have, using our inductive hypothesis

$$\begin{aligned}
||B^{\ell+1}V - B^{\ell+1}V'|| &= ||B^1(B^\ell V) - B^1(B^\ell V')|| \\
&= ||(B^\ell V) - (B^\ell V')|| \\
&\leq \gamma^1||(B^\ell V) - (B^\ell V')|| \\
&\leq \gamma^1\gamma^\ell||V - V'|| \\
&\gamma^{\ell+1}||V - V'||,
\end{aligned}$$

completing the proof.

(g) Prove that $||V^* - V_{n+1}|| \leq \varepsilon/2$. [7pts]

Hints: Note that $||V^* - V_{n+1}|| = ||V^* + V_{n+2} - V_{n+2} - V_{n+1}||$ and you can repeatedly apply this trick. It may also be useful to leverage part (f) and recall that V^* is the fixed point of the contraction B.

Iteratively and infinitely insert terms of increasing index, and then apply triangle inequality and previous findings, i.e.

$$\begin{aligned}
||V^* - V_{n+1}|| &= ||V^* + V_{n+2} - V_{n+2} - V_{n+1}|| \\
&= ||V^* + V_{n+3} - V_{n+3} + V_{n+2} - V_{n+2} - V_{n+1}|| \\
&\vdots \\
&= \left|\left| \sum_{i=1}^{\infty} V_{n+i+1} - V_{n+i} \right|\right| \\
&\leq \sum_{i=1}^{\infty} ||V_{n+i+1} - V_{n+i}|| \\
&= \sum_{i=1}^{\infty} ||B^i V_{n+1} - B^i V_n|| \\
&= \sum_{i=1}^{\infty} ||B^i V_{n+1} - B^i V_n|| \\
&\leq \sum_{i=1}^{\infty} \gamma^i ||V_{n+1} - V_n|| \\
&\leq \frac{\varepsilon(1-\gamma)}{2\gamma} \sum_{i=1}^{\infty} \gamma^i \\
&\leq \frac{\varepsilon(1-\gamma)}{2\gamma} \cdot \frac{\gamma}{(1-\gamma)} \\
&= \frac{\varepsilon}{2},
\end{aligned}$$

with the last move a standard geometric series (as $\gamma \in (0, 1)$. This completes the proof.

(h) Use the results from parts (e) and (g), to show that $||V_\pi - V^*|| \leq \varepsilon$ [2 pts]

Using results above and the triangle inequality,

$$\begin{aligned}
||V_\pi - V^*|| &= ||V_\pi - V_{n+1} + V_{n+1} - V^*|| \\
&\leq ||V_\pi - V_{n+1}|| + ||V_{n+1} - V^*|| \\
&= ||V_\pi - V_{n+1}|| + ||V^* - V_{n+1}|| \\
&= \frac{\varepsilon}{2} + \frac{\varepsilon}{2} \\
&= \varepsilon.
\end{aligned}$$

3 Frozen Lake MDP [25 pts]

Now you will implement value iteration and policy iteration for the Frozen Lake environment from OpenAI Gym. We have provided custom versions of this environment in the starter code.

(a) **(coding)** Read through `vi_and_pi.py` and implement `policy_evaluation`, `policy_improvement` and `policy_iteration`. The stopping tolerance (defined as $\max_s |V_{old}(s) - V_{new}(s)|$) is tol $= 10^{-3}$. Use $\gamma = 0.9$. Return the optimal value function and the optimal policy. [10pts]

(b) **(coding)** Implement `value_iteration` in `vi_and_pi.py`. The stopping tolerance is tol $= 10^{-3}$. Use $\gamma = 0.9$. Return the optimal value function and the optimal policy. [10 pts]

(c) **(written)** Run both methods on the Deterministic-4x4-FrozenLake-v0 and

Stochastic-4x4-FrozenLake-v0 environments. In the second environment, the dynamics of the world are stochastic. How does stochasticity affect the number of iterations required, and the resulting policy? [5 pts]

Running the stochastic environment multiple times, the stochasticity seems to impact things in the following ways:

(1) it may change the policy prescription. As PI and VI would have to sum/weight/take expectations over different transition probabilities for a given action/state (as opposed to the deterministic moves of before), calculations, and in turn, optimal policies in a given position may differ. In other words, the stochasticity may change what is optimal for a given state. (2) It may increase the number of iterations, both in policy iteration and value iteration. When attempting to move states according to a policy, the stochasticity may reject a move and send the player backwards, forcing them to regain progress already established. In other words, the stochasticity introduces more 2 steps forward, one step back moves that require more iterations to overcome. (3) The stochasticity may lead players to fall in the lake. Even if value iteration and policy iteration converge to what is an optimal policy, the player may still get an unlucky random draw and fall in the lake, through no fault of their own. That's the "slippery" nature of the lake.