CS 234 Winter 2021
Assignment 1
Due: January 22 at 6:00 pm (PST)

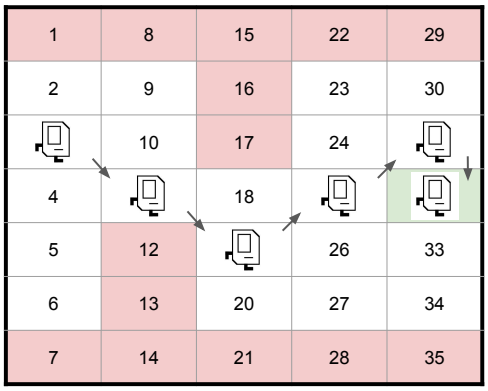
For submission instructions please refer to website For all problems, if you use an existing result from either the literature or a textbook to solve the exercise, you need to cite the source.

1 Flappy Karel MDP [25 pts]

There is a hot new mobile game on the market called Flappy Karel, where Karel the robot must dodge the red pillars of doom and flap its way to the green pasture. Consider the following 2 grid environments (Flappy World 1 and Flappy World 2). Starting from any unshaded square, Karel can either move right & up, or right & down (e.g from state 4 you can move to state 10 or 12, think checkers). Actions are deterministic and always succeed unless they will cause Karel to run into a wall. The thicker edges indicate walls, and attempting to move in the direction of a wall results in falling down one square (e.g. going in any direction from state 30 leads to falling into state 31). A successful run by Karel in Flappy World 1 is shown in Figure 1b. Taking any action from the green target squares (no. 32) earns a reward of r_g and ends the episode. Taking any action from the red squares of doom (no. 1, 7, 8, 12, 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 in both worlds.

1	8	15	22	29
2	9	16	23	30
3	10	17	24	31
4	11	18	25	32
5	12	19	26	33
6	13	20	27	34
7	14	21	28	35

(a) Flappy World 1



(b) A successful run by Karel in Flappy World 1

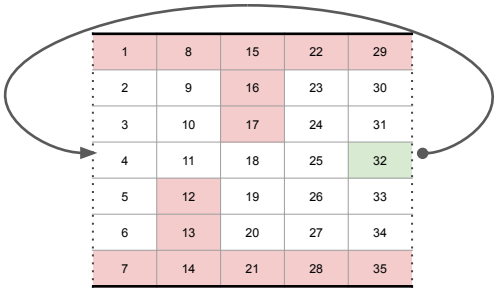
Figure 1

-4: 2->8
-1: 2->10->18->26->32
0: wander around until reach 32
1: wander around (longer path) to get more reward

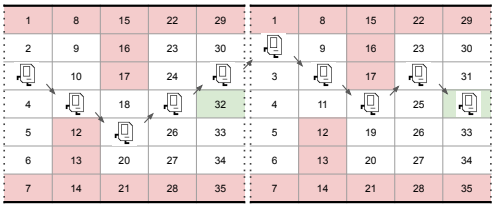
(a) Let $r_s \in \{-4, -1, 0, 1\}$. Starting in **square 2**, for each of the possible values of r_s briefly explain what the optimal policy would be in Flappy World 1. In each case is the optimal policy unique and does the optimal policy depend on the value of the discount factor γ? Explain your answer. [5 pts]

(b) What value of r_s would cause the optimal policy to return the shortest path to the green target square? Using this value of r_s find the optimal value function for each square in Flappy world 1. What is the optimal action from square 27? [5 pts]

Now consider Flappy world 2. It is the same as Flappy world 1, except there are no walls on the right and left sides. Going past the right end of flappy world 2 simply loops you to left hand side. Take a look at Figure 1b for a successful run by Karel in Flappy World 2.



(a) Flappy World 2



(b) A successful run by Karel in Flappy World 2

Figure 2

(c) Let $r_s \in \{-4, -1, 0, 1\}$. Starting in **square 3**, for each of the possible values of r_s briefly explain what the optimal policy would be in Flappy World 2. Using the value of r_s, that would cause the optimal policy to return the shortest path to the green target square, find the optimal value function for each square in Flappy world 2. What is the optimal action from square 27? [5 pts]

(d) Consider a general MDP with rewards, and transitions. Consider a discount factor of γ. For this case assume that the horizon is infinite (so there is no termination). A policy π in this MDP induces a value function V^π (lets refer to this as V^π_{old}). Now suppose we have the same MDP where all rewards have a constant c added to them and then have been scaled by a constant a (i.e. $r_{new} = a(c + r_{old})$). Can you come up with an expression for the new value function V^π induced by π in this second MDP in terms of V^π_{old}, c, a, and γ? [5 pts]

(e) Can scaling all the rewards by a fixed amount change the optimal policy of a MDP? If so, describe how different ranges of the constant a (where $r_{new} = a * (r_{old})$) would change the optimal policy of the MDP from part (c). [5 pts]

2 Applications of the Performance Difference Lemma [20pts]

The purpose of this exercise is to get familiar on how to compare the value of different policies, π_1 and π_2, on a fixed horizon MDP. A fixed horizon MDP is an MDP where the agent's state is reset after H timesteps; H is called the *horizon* of the MDP. There is no discount (i.e., $\gamma = 1$) and policies are allowed to be non-stationary, i.e., the action identified by a policy depends on the timestep in addition to the state. Let $x_t \sim \pi$ denote the distribution over states at timestep t (for $1 \leq t \leq H$) upon following policy π and $V_t^\pi(x_t)$ denote the value function of policy π in state x_t and timestep t, and $Q_t^\pi(x_t, a)$ denote the corresponding Q value associated to action a. As a clarifying example, we denote $\mathbb{E}_{x_t \sim \pi_1} V(x_t)$ to represent the average value of the value function $V(\cdot)$ over the states at timestep t encountered upon following policy π_1. The following equality is called *performance difference lemma* :

$$V_1^{\pi_1}(x_1) - V_1^{\pi_2}(x_1) = \sum_{t=1}^{H} \mathbb{E}_{x_t \sim \pi_2} \Big(Q_t^{\pi_1}(x_t, \pi_1(x_t, t)) - Q_t^{\pi_1}(x_t, \pi_2(x_t, t)) \Big)$$

Intuition: The above expression can be interpreted in the following way. For concreteness, assume that π_1 is the better policy, i.e., achieving $V_1^{\pi_1}(x_1) \geq V_1^{\pi_2}(x_1)$. Suppose you're following policy π_2 and you are at timestep t in state x_t. You have the option to follow π_1 (the better policy) until the end of the episode, totalling $Q_t^{\pi_1}(x_t, \pi_1(x_t, t))$ return from the current state-timestep; or you have the option to follow π_2 for one timestep and then follow π_1 instead until the end of the episode (you can follow many other policies of course). This would give you a "loss" of $Q_t^{\pi_1}(x_t, \pi_1(x_t, t)) - Q_t^{\pi_1}(x_t, \pi_2(x_t, t))$ that originates from following the worse policy π_2 instead of π_1 in that timestep. Then the equation above means that the value difference of the two policies is the sum of all the losses induced by following the suboptimal policy for every timestep, weighted by the expected trajectory of the policy you're following.

Question You will use the performance difference lemma to solve this problem. Consider an MDP where the state space $\mathcal{S}$ is partitioned into two sets of states $\mathcal{S}^+$ and its complement $\overline{\mathcal{S}}^+$.

$$\mathcal{S} = \mathcal{S}^+ \cup \overline{\mathcal{S}}^+$$
$$\mathcal{S}^+ \cap \overline{\mathcal{S}}^+ = \emptyset.$$

In every state $s \in \mathcal{S}^+$ there exists an action a^+ that leads to the same state with probability 1 and gives a unitary reward:

$$p(s_{t+1} = s \mid s_t = s, a_t = a^+) = 1, \qquad p(st + 1 \neq s \mid s_t = s, a_t = a^+) = 0.$$

The reward function is always positive. In $\mathcal{S}^+$ the reward function equals 1 upon playing a^+ and H upon playing any action $a \neq a^+$. Therefore in $\mathcal{S}^+$

$$r(s, a^+) = 1, \quad r(s, a) = H, \quad a \neq a^+$$

Conversely, in any state $s \notin S^+$, the reward function is in $[0, 1]$ ($\forall s \notin S^+ \quad \forall a \;\; r(s, a) \in [0, 1]$).

Consider a policy π and define a policy π^+ that takes action a^+ in any state $\mathcal{S}^+$ and is otherwise equal to π:

$$\pi^+(s) = a^+ \text{ if } s \in \mathcal{S}^+, \qquad \pi^+(s) = \pi(s) \text{ if } s \notin \mathcal{S}^+$$

Intuitively, π accumulates higher return than π^+: in any state in $\mathcal{S}^+$ the policy π^+ chooses to take a unitary reward forever instead of a reward of H and then maybe more. Using the performance difference lemma show that at any state s_0

$$V_1^{\pi}(s_0) \geq V_1^{\pi^+}(s_0).$$

3 Nonstationary Discount Factor γ [30 pts]

In this problem you will consider a variable discount factor γ. In lecture 2, we proved that the Bellman backup is a contraction for $\gamma < 1$ in the infinity norm.

In this problem we consider having a non-stationary discount factor and assume you want to run K iterations of value iterations. Let V_K and V'_K be any two arbitrary initial value functions (at timestep K). The time-dependent Bellman backup operator B_k is defined as

$$V_{k-1} \stackrel{def}{=} B_k V_k = \max_a [R(s,a) + \gamma_k \sum_{s' \in S} p(s'|s,a) V_k(s')]$$

where

$$\gamma_k = 1 - \frac{1}{k+1}$$

Notice that the value function index is decreasing: $K, K-1, \ldots, 2, 1$

10pt Similarly to what you've done in class, show that the Bellman operator with non-stationary discount factor at time step k is still a contraction, i.e.,

$$\|B_k V - B_k V'\|_\infty \leq \gamma_k \|V - V'\|_\infty$$

10pt Using the above inequality prove that

$$\|B_1 B_2 \cdots B_K V_K - B_1 B_2 \cdots B_K V'_K\|_\infty \leq \gamma_1 \gamma_2 \cdots \gamma_k \|V_K - V'_K\|_\infty$$

10pt Unfortunately $\gamma_k \approx 1$ when k is large so you cannot conclude that the convergence occurs exponentially fast. However, the error still shrinks: show that

$$\gamma_1 \gamma_2 \cdots \gamma_K \leq \frac{1}{K+1}$$

which allows you to write

$$\|B_1 B_2 \cdots B_K V_K - B_1 B_2 \cdots B_K V'_K\|_\infty \leq \frac{1}{K+1} \|V_K - V'_K\|_\infty$$

and ensure convergence, albeit at a slower rate.

4 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]