CS 188 Spring 2019 Final Review MDPs and RL Solutions

Q1. MDPs - Farmland

In the game FARMLAND, players alternate taking turns drawing a card from one of two piles, PIG and COW. PIG cards have equal probability of being 3 points or 1 point, and COW cards are always worth 2 points. Players are trying to be the first to reach 5 points or more. We are designing an agent to play FARMLAND.

We will use a modified MDP to come up with a policy to play this game. States will be represented as tuples (x, y) where x is our score and y is the opponent's score. The value $V(x, y)$ is an estimate of the probability that we will win at the state (x, y) **when it is our turn to play** and both players are playing optimally. Unless otherwise stated, assume both players play optimally.

First, suppose we work out by hand V^*, the table of actual probabilities.

		Opponent				
		0	1	2	3	4
	0	0.75	0.5	0.5	0	0
	1	1	1	0.5	0	0
You	2	1	1	0.75	0.5	0.5
	3	1	1	1	1	1
	4	1	1	1	1	1

According to this table, $V^*(1, 2) = 0.5$, so with both players playing optimally, the probability that we will win if our score is 1, the opponent's score is 2, and it is our turn to play is 0.5.

(a) At the beginning of the game, would you choose to go first or second? Justify your answer using the table.

You should choose to go first. Since $V^*(0, 0) = 0.75$, if it is your turn and the scores are both 0, the probability that you will win is 0.75.

(b) If our current state is (x, y) (so our score is x and the opponent's score is y) but it is **the opponent's turn to play**, what is the probability that we will win if both players play optimally **in terms of V***?

$$1 - V^*(y, x)$$

(c) As FARMLAND is a very simple game, you quickly grow tired of playing it. You decide to buy the FARMLAND expansion, BOVINE BONANZA, which adds loads of exciting cards to the COW pile! Of course, this changes the transition function for our MDP, so the table V^* above is no longer correct. We need to come up with an update equation that will ultimately make V_∞ converge on the actual probabilities that we will win.

You are given the transition function $T((x, y), a, (x', y))$ and the reward function $R((x, y), a, (x', y))$. The transition function $T((x, y), a, (x', y))$ is the probability of transitioning from state (x, y) to state (x', y) when action a is taken (i.e. the probability that the card drawn gives $x' - x$ points).

Since we are only trying to find the probability of winning and we don't care about the margin of victory, the reward function $R((x, y), a, (x', y))$ is 1 whenever (x', y) is a winning state and 0 everywhere else. As in normal value iteration, all values will be initialized to 0 (i.e. $V(x, y) = 0$ for all states (x, y)).

Write an update equation for $V_{k+1}(x, y)$ in terms of T, R and V_k.
Hint: you will need to use your answer from part b.

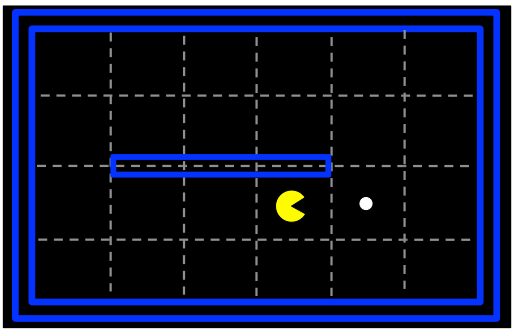
$$V_{k+1}(x, y) \leftarrow \max_a \sum_{x'} T((x, y), a, (x', y))[R((x, y), a, (x', y)) + (1 - V_k(y, x'))]$$

Q2. Solving Search Problems with MDPs

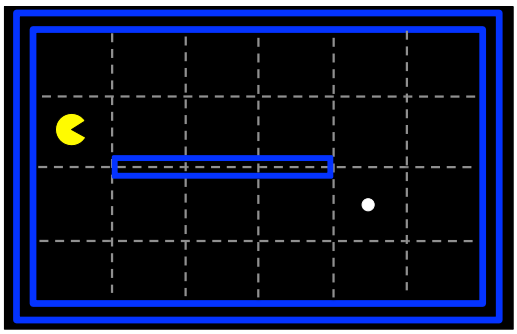
The following parts consider a Pacman agent in a deterministic environment. A goal state is reached when there are no remaining food pellets on the board. Pacman's available actions are $\{N, S, E, W\}$, but Pacman **can not** move into a wall. Whenever Pacman eats a food pellet he receives a reward of $+1$.

Assume that pacman eats a food pellet as soon as he occupies the location of the food pellet—i.e., the reward is received for the transition into the square with the food pellet.

Consider the particular Pacman board states shown below. Throughout this problem assume that $V_0(s) = 0$ for all states, s. Let the discount factor, $\gamma = 1$.



State A



State B

(a) What is the optimal value of state A, $V^*(A)$?

1

(b) What is the optimal value of state B, $V^*(B)$?

1

The reason the answers are the same for both (b) and (a) is that there is no penalty for existing. With a discount factor of 1, eating the food at any future step is just as valuable as eating it on the next step. An optimal policy will definitely find the food, so the optimal value of any state is always 1.

(c) At what iteration, k, will $V_k(B)$ first be non-zero?

5

The value function at iteration k is equivalent to the maximum reward possible within k steps of the state in question, B. Since the food pellet is exactly 5 steps away from Pacman in state B, $V_5(B) = 1$ and $V_{K<5}(B) = 0$.

(d) How do the optimal q-state values of moving W and E from state A compare? (*choose one*)

○ $Q^*(A, W) > Q^*(A, E)$ ○ $Q^*(A, W) < Q^*(A, E)$ ● $Q^*(A, W) = Q^*(A, E)$

Once again, since $\gamma = 1$, the optimal value of every state is the same, since the optimal policy will eventually eat the food.

(e) If we use this MDP formulation, is the policy found guaranteed to produce the shortest path from pacman's starting position to the food pellet? If not, how could you modify the MDP formulation to guarantee that the optimal policy found will produce the shortest path from pacman's starting position to the food pellet?

No. The Q-values for going $West$ and $East$ from state A are equal so there is no preference given to the

shortest path to the goal state. Adding a negative living reward (example: -1 for every time step) will help differentiate between two paths of different lengths. Setting $\gamma < 1$ will make rewards seen in the future worth less than those seen right now, incentivizing Pacman to arrive at the goal as early as possible.