CS 188
Fall 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. Spinaroo

A casino considers adding the game Spinaroo to their collection, but needs you to analyze it before releasing on their floor. The game starts by the dealer rolling a 4-sided die, which can take on values $\{1, 2, 3, 4\}$ with equal probability. You get to observe this rolled value, D (for dealer). You are then given a separate 2-sided die, which can take on values $\{1, 2\}$ with equal probability. You are initially forced to roll this die once and observe its value G (for gambler). At this point, you can choose whether to continue rolling or to stop. Each time you roll the die, the observed value gets added to the cumulative sum G. Once you stop, the game ends. If the cumulative sum $G < D$, you lose 1 dollar. If $G = D$, you neither win nor lose money. If $D < G < 5$, you win 1 dollar. If $G \geq 5$, you lose 1 dollar.

You decide to model this game via a Markov Decision Process (MDP). You model the states as tuples $(d, g) \in \{1, 2, 3, 4\} \times \{1, 2, 3, 4, Bust\}$, where d denotes the dealer's roll and g denotes the current cumulative sum. In particular, we set g to $Bust$ when it is 5 or higher. After a players first forced roll, their available actions are *Roll* and *Stop*. The reward, the amount of money they win, is awarded once they *Stop*, transitioning them to the *End* state. The discount factor is 1.

(a) You first consider policy iteration in solving this problem. The initial policy π is in the table below. For example, the initial policy prescribes *Roll* in the state $(a, b) = (3, 2)$.

	$d = 1$	$d = 2$	$d = 3$	$d = 4$
$g = 1$	*Roll*	*Roll*	*Stop*	*Roll*
$g = 2$	*Stop*	*Stop*	*Stop*	*Stop*
$g = 3$	*Stop*	*Stop*	*Roll*	*Roll*
$g = 4$	*Stop*	*Stop*	*Stop*	*Stop*

Fill in the following table, denoting V^{π}. Some values have been filled in for you.

	$d = 1$	$d = 2$	$d = 3$	$d = 4$
$g = 1$	1	$\frac{1}{2}$	-1	$-\frac{3}{4}$
$g = 2$	1	0	-1	-1
$g = 3$	1	1	0	$-\frac{1}{2}$
$g = 4$	1	1	1	0

(b) At some point of time during policy iteration, you notice that V^{π} is as follows:

	$d=1$	$d=2$	$d=3$	$d=4$
$g=1$	1	1	$\frac{1}{4}$	$-\frac{3}{8}$
$g=2$	1	1	$\frac{1}{2}$	$-\frac{1}{4}$
$g=3$	1	1	0	$-\frac{1}{2}$
$g=4$	1	1	1	0

where π was:

	$d=1$	$d=2$	$d=3$	$d=4$
$g=1$	*Roll*	*Roll*	*Roll*	*Roll*
$g=2$	*Stop*	*Roll*	*Roll*	*Roll*
$g=3$	*Stop*	*Stop*	*Roll*	*Roll*
$g=4$	*Stop*	*Stop*	*Stop*	*Stop*

Perform a policy improvement step to extract policy π'. In the case where both Roll and Stop are acceptable updates, write Roll/Stop. Parts of the policy have been filled in:

	$d=1$	$d=2$	$d=3$	$d=4$
$g=1$	*Roll*	*Roll*	*Roll*	*Roll*
$g=2$	*Stop*	*Roll*	*Roll*	*Roll*
$g=3$	*Stop*	*Stop*	*Roll/Stop*	*Roll*
$g=4$	*Stop*	*Stop*	*Stop*	*Stop*

(c) Is the policy, π, given from the previous part optimal? Note that we're asking about the optimality of policy π and not π'. From the information we have, can we deduce whether policy iteration has converged? Briefly explain your reasoning.

Policy iteration has converged, since the set of policies we consider updating contains π. π is optimal, since policy iteration converges if and only if policy iteration yields an optimal policy.

(d) Spinaroo is released to the floor, and it's doing very well! The casino is concerned that players might determine the optimal policy, so they decide to use weighted dice (for both the 2 and 4-sided dice), where the weights are unknown.

The casino wants you to determine an optimal policy for this Spinaroo variant. You decide to use Q-Learning to solve this problem, where states, actions, and rewards are modeled as in the previous questions. Select all strategies that will guarantee that a certain player learns the optimal Q-values.

☐ Strategy 1: During learning, the placer acts according to the optimal policy π: namely, the policy where $\pi(s) = \text{argmax}_a Q(s, a)$. Learning continues until convergence.

☐ Strategy 2: During learning, the player acts according to the pessimal policy π: namely, the policy where $\pi(s) = \text{argmin}_a Q(s, a)$. Learning continues until convergence.

■ Strategy 3: During learning, the player chooses *Roll* and *Stop* at random. Learning continues until convergence.

☐ Strategy 4: During learning, the player chooses *Roll* and *Stop* at random. Learning continues until each state-action pair has been seen at least 20 times.

■ Strategy 5: During learning, in a state s, the player chooses the action $a \in \{Roll, Stop\}$ that the placer has chosen least often in s, breaking ties randomly. Learning continues until convergence.

(e) Your manager is proud of being an indecisive person. As such, you decide to impress your manager by devising indecisive learning strategies. Each choice X, Y below corresponds to the following strategy: when asked to take an action, choose the action prescribed by strategy X with probability $0 < \epsilon < 1$ and the action prescribed by strategy Y with probability $1 - \epsilon$. Refer to the previous part for the names of the strategies. Which of the following indecisive strategies will lead to learning the optimal Q-values? Note: in the case where strategy 4 is used, learning continues until each state-action pair has been seen at least 20 times, and not until convergence.

Hint: consider the number of actions that are available.

■ $1, 2$
■ $1, 3$
☐ $1, 4$
■ $1, 5$
■ $2, 3$
☐ $2, 4$
■ $2, 5$
■ $3, 5$
☐ $3, 4$
☐ $4, 5$