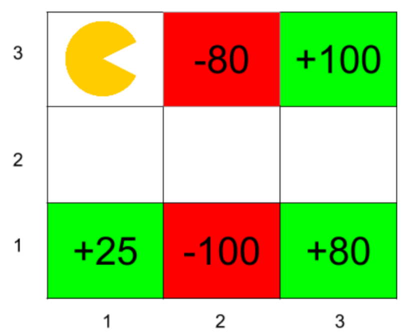
CS 6300 **HW6: Q Learning and Functional Approximation** **Due March 7, 2017**

Please use the LaTeX template to produce your writeups. See the Homework Assignments page on the class website for details. Hand in at: `https://webhandin.eng.utah.edu/index.php`.

1 Approximate Q-Learning

Consider the grid-world given below and Pacman who is trying to learn the optimal policy. If an action results in landing into one of the shaded states the corresponding reward is awarded during that transition. All shaded states are terminal states, i.e., the MDP terminates once arrived in a shaded state. The other states have the *North, East, South, West* actions available, which deterministically move Pacman to the corresponding neighboring state (or have Pacman stay in place if the action tries to move out of the grad). Assume the discount factor $\gamma = 0.5$ and the Q-learning rate $\alpha = 0.5$ for all calculations. Pacman starts in state (1, 3).



1. What is the value of the optimal value function V^* at the following states:

State	**Optimal value**
$V^*(3,2)$	100
$V^*(2,2)$	50
$V^*(1,3)$	12.5

We want to calculate the optimal values from each of the states *onwards*, **NOT** up until that point. The question states that the award is given *during transition into the state*, so we do not need to discount to get the award. Also, as we want to chose the maximum, we get the following:

- $(3,2)$: Maximize to 100 over 80 since we want optimal
- $(2,2)$: Go to 100 square, with a $1/2$ discount, $0 + \gamma(100) = 50$ to maximize
- $(1,3)$: Going to either 25 or 100 has the discounted reward of 12.5

Episode 1	Episode 2	Episode 3
(1,3), S, (1,2), 0	(1,3), S, (1,2), 0	(1,3), S, (1,2), 0
(1,2), E, (2,2), 0	(1,2), E, (2,2), 0	(1,2), E, (2,2), 0
(2,2), S, (2,1), -100	(2,2), E, (3,2), 0	(2,2), E, (3,2), 0
	(3,2), N, (3,3), +100	(3,2), S, (3,1), +80

2. The agent starts from the top left corner and you are given the following episodes from runs of the agent through this grid-world. Each line in an Episode is a tuple containing (s, a, s', r).

   Using Q-Learning updates, what are the following Q-values after the above three episodes:

Q State	**Value**
$Q((3,2),N)$	50
$Q((1,2),S)$	0
$Q((2,2),E)$	12.5

   The Q-values can be found by the learning updates with the equation

$$Q(s,a) = (1-\alpha)Q(s,a) + \alpha\left[R(s,a,s') + \gamma \max_{a'} Q(s',a')\right]$$

   We can do it for the first example, where we get $R(s,a,s')$ as the reward is given on *transition*, giving

$$Q((3,2),N) = 0.5(0) + 0.5\left[100 + 0.5\max_{a'}(0)\right] = 50$$

3. Consider a feature based representation of the Q-value function:

$$Q_f(s,a) = w_1 f_1(s) + w_2 f_2(s) + w_3 f_3(a)$$

   $f_1(s)$: The x coordinate of the state $f_2(s)$: The y coordinate of the state

$$f_3(N) = 1,\ f_3(S) = 2,\ f_3(E) = 3,\ f_3(W) = 4$$

   (a) Given that all w_i are initially 0, what are their values after the first episode using approximate Q-learning weight updates.

Weight	**Value**
w_1	-100
w_2	-100
w_3	-100

Using the equation for the Q-learning,

$$w_i = w_i + \alpha \left[R(s,a,s') + \gamma \max_{a'} Q(s',a') - Q(s,a) \right] f_i(s,a)$$

In calculating this for the first weight, we get

$$\begin{aligned} w_1 &= 0 + 0.5\,[-100 + 0.5(0)]\, f_1\,((2,2)) \\ &= 0 + 0.5\,[-100 + 0.5(0)]\, 2 = -100 \end{aligned}$$

which when plugging in the values, we get -100 for each of the weights for the first episode.

(b) Assume the weight vector w is equal to $(1,1,1)$. What is the action prescribed by the Q-function in state $(2,2)$?

At $(2,2)$, the best action to maximize the value of $Q((2,2),a)$ is computed and is maximum for `west` which gives $2+2+4$ which is a value of 8.

2 Functional Approximation

In this question, you will play a simplied version of blackjack where the deck is infinite and the dealer always has a fixed count of 15. The deck contains cards 2 through 10, J, Q, K, and A, each of which is equally likely to appear when a card is drawn. Each number card is worth the number of points shown on it, the cards J, Q, and K are worth 10 points, and A is worth 11. At each turn, you may either *hit* or *stay*.

- If you choose to *hit*, you receive no immediate reward and are dealt an additional card.
- If you stay, you receive a reward of 0 if your current point total is exactly 15, +10 if it is higher than 15 but not higher than 21, and -10 otherwise (i.e., lower than 15 or larger than 21).
- After taking the *stay* action, the game enters a terminal state *end* and ends.
- A total of 22 or higher is refered to as a *bust*; from a *bust*, you can only choose the action *stay*.

As your state space you take the set $\{0, 2, \ldots, 21, bust, end\}$ indicating point totals.

1. Suppose you have performed k iterations of value iteration. Compute $V_{k+1}(12)$ given the partial table below for $V_k(s)$. Give your answer in terms of the discount as a variable.

s	$V_k(s)$
13	2
14	10
15	10
16	10
17	10
18	10
19	10
20	10
21	10
bust	-10
end	0

Out of all of the cards, $(2-9)$ will take us to `return` and $(10-A)$ will take us to `bust`. This means that $V_{k+1}(12)$ is

$$V_{k+1}(12) = \frac{1}{13}\left[8(10\gamma) + 5(-10\gamma)\right]$$
$$= \frac{30}{13}\gamma$$

2. You suspect that the cards do not actually appear with equal probability, and decide to use Q-learning instead of value iteration. Given the partial table of initial Q-values below left, fill in the partial table of Q-values on the right after the episode center below occurs. Assume $\alpha = 0.5$ and $\gamma = 1$. The initial portion of the episode has been omitted. Leave blank any values which Q-learning does not update.

s	a	$Q(s,a)$
19	hit	-2
19	stay	5
20	hit	-4
20	stay	7
21	hit	-6
21	stay	8
bust	stay	-8

s	a	r	s'
19	hit	0	21
21	hit	0	bust
bust	stay	-10	end

s	a	$Q(s,a)$
19	hit	3
19	stay	
20	hit	
20	stay	
21	hit	-7
21	stay	
bust	stay	-9

These are calculated with the Q-learning equation, where the example for the first can be seen below

$$
\begin{aligned}
Q(19, hit) &= \frac{1}{2}(-2) + \frac{1}{2}\left[0 + 1 \cdot \max(8, -6)\right] \\
&= 3
\end{aligned}
$$

Where the others follow a similar nature.

3. Unhappy with your experience with basic Q-learning, you decide to featurize your Q-values. Consider the two feature functions:

$$
f_1(s,a) = \begin{cases} 0 & a = stay \\ +1 & a = hit, s \geq 15 \\ -1 & a = hit, s < 15 \end{cases} \quad \text{and} \quad f_2(s,a) = \begin{cases} 0 & a = stay \\ +1 & a = hit, s \geq 18 \\ -1 & a = hit, s < 18 \end{cases}
$$

Which of the following partial policy tables may be represented by the featurized Q-values unambiguously (without ties)?

(a)

s	$\pi(s)$
14	hit
15	hit
16	hit
17	hit
18	hit
19	hit

(b)

s	$\pi(s)$
14	stay
15	hit
16	hit
17	hit
18	stay
19	stay

(c)

s	$\pi(s)$
14	hit
15	hit
16	hit
17	hit
18	stay
19	stay

(d)

s	$\pi(s)$
14	hit
15	hit
16	hit
17	hit
18	hit
19	stay

(e)

s	$\pi(s)$
14	hit
15	hit
16	hit
17	stay
18	hit
19	stay

The featurized Q-values can be calculated as $\omega_1 f_1(s,a) + \omega_2 f_2(s,a)$, where ω_1 and ω_2 are the weights for the features given above. Therefore, we need to calculate the Q-values to see if they hold. We get only (c) as not having to break ties.

For example, we have in the table for (a)

$Q(s,a)$	hit	$stay$
14	$-\omega_1 - \omega_2$	0
15	$\omega_1 - \omega_2$	0
16	$\omega_1 - \omega_2$	0
17	$\omega_1 - \omega_2$	0
18	$\omega_1 + \omega_2$	0
19	$\omega_1 + \omega_2$	0

From here, we then need to go through and see if the policy has contradictions. This gives

$$\begin{aligned} Q(14, hit) = -\omega_1 - \omega_2 > Q(14, stay) = 0 \\ \Rightarrow -\omega_1 - \omega_2 > 0 \end{aligned}$$

Howerver, we get for $s = 18$

$$\begin{aligned} Q(18, hit) = \omega_1 + \omega_2 > Q(18, stay) = 0 \\ \Rightarrow \omega_1 + \omega_2 > 0 \end{aligned}$$

This leads to (a) failing, as it does for every other policy besides (c).