AI Homework III

Student Number: 1642721

Date: April 25, 2016

Question 1:

a) There are 2 distinct policies: going left and going right.

b) $(\boldsymbol{Value\ going\ left})$

$$\sum_{t=1} \gamma^t R(s_t)\, U(s')$$

$$= 0 + 0.9(0) + 0.9^2(0) + 0.9^3(5)$$
$$= 0 + 0 + 0 + 3.645$$
$$= 3.645 * 0.2$$
$$= 0.729$$

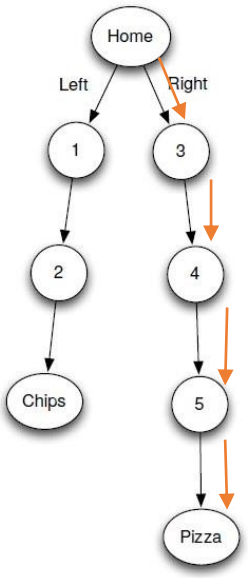
$(\boldsymbol{Value\ going\ right})$

$$\sum_{t=1} \gamma^t R(s_t)\, U(s')$$

$$= 0 + 0.9(0) + 0.9^2(0) + 0.9^3(0) + 0.9^4(10)$$
$$= 0 + 0 + 0 + 0 + 6.561$$
$$= 6.561 * 0.8$$
$$= 5.2488$$

$$U(home) = U(home, left) + U(home, right)$$
$$= 0.729 + 5.2488$$
$$= 5.9778$$

c) The optimal policy is to go right at a value of 5.2488 (calculated above)



d) $\gamma^3(5) < \gamma^4(10)$
$= \gamma < 0.5$

Question 2:

Let S = {p)

Let A = {left, right, up, down}

Bolded answers are the max for the noted action

$Q(p, left) = -4 + 0.9(0.5 * 0) + 0.9(0.25 * 0) + 0.9(0.25 * -100) = \mathbf{-26.5}$

$Q(p, left) = -4 + 0.9(0.5 * -26.5) + 0.9(0.25 * -26.5) + 0.9(0.25 * -100) = -44.3875$

$Q(p, left) = -4 + 0.9(0.5 * -44.3875) + 0.9(0.25 * -44.3875) + 0.9(0.25 * -100)$
$= -56.4615625$

$Q(p, left) = -4 + 0.9(0.5 * -56.4615625) + 0.9(0.25 * -56.4615625) + 0.9(0.25 * -100)$
$= -64.61155469$

$Q(p, left) = -4 + 0.9(0.5 * -64.61155469) + 0.9(0.25 * -64.61155469) + 0.9(0.25 * -100)$
$= -70.11279942$

$Q(p, left) = -4 + 0.9(0.5 * -70.11279942) + 0.9(0.25 * -70.11279942) + 0.9(0.25 * -100)$
$= -73.82613961$

$Q(p, left) = -4 + 0.9(0.5 * -73.82613961) + 0.9(0.25 * -73.82613961) + 0.9(0.25 * -100)$
$= -76.33264423$

$Q(p, left) = -4 + 0.9(0.5 * -76.33264423) + 0.9(0.25 * -76.33264423) + 0.9(0.25 * -100)$
$= -78.02453485$

$$Q(p, left) = -4 + 0.9(0.5 * -78.02453485) + 0.9(0.25 * -78.02453485) + 0.9(0.25 * -100)$$
$$= -79.16656102$$

$$Q(p, left) = -4 + 0.9(0.5 * -79.16656102) + 0.9(0.25 * -79.16656102) + 0.9(0.25 * -100)$$
$$= -79.93742869$$

$$Q(p, right) = -4 + 0.9(0.5 * 100) + 0.9(0.25 * 0) + 0.9(0.25 * -100) = 18.5$$

$$Q(p, right) = -4 + 0.9(0.5 * 100) + 0.9(0.25 * 18.5) + 0.9(0.25 * -100) = 22.6625$$

$$Q(p, right) = -4 + 0.9(0.5 * 100) + 0.9(0.25 * 22.6625) + 0.9(0.25 * -100) = 23.5990625$$

$$Q(p, right) = -4 + 0.9(0.5 * 100) + 0.9(0.25 * 23.5996025) + 0.9(0.25 * -100) = 23.80978906$$

$$Q(p, right) = -4 + 0.9(0.5 * 100) + 0.9(0.25 * 23.80978906) + 0.9(0.25 * -100)$$
$$= \mathbf{23.85720254}$$

$$Q(p, up) = -4 + 0.9(0.5 * 0) + 0.9(0.25 * 100) + 0.9(0.25 * 0) = 18.5$$

$$Q(p, up) = -4 + 0.9(0.5 * 18.5) + 0.9(0.25 * 100) + 0.9(0.25 * 18.5) = 30.9875$$

$$Q(p, up) = -4 + 0.9(0.5 * 30.9875) + 0.9(0.25 * 100) + 0.9(0.25 * 30.9875) = 39.4165625$$

$$Q(p, up) = -4 + 0.9(0.5 * 39.4165625) + 0.9(0.25 * 100) + 0.9(0.25 * 39.4165625)$$
$$= 45.10617969$$

$$Q(p, up) = -4 + 0.9(0.5 * 45.10617969) + 0.9(0.25 * 100) + 0.9(0.25 * 45.10617969)$$
$$= 48.94667129$$

$$Q(p, up) = -4 + 0.9(0.5 * 48.94667129) + 0.9(0.25 * 100) + 0.9(0.25 * 48.94667129)$$
$$= 51.53900312$$

$$Q(p, up) = -4 + 0.9(0.5 * 51.53900312) + 0.9(0.25 * 100) + 0.9(0.25 * 51.53900312)$$
$$= 53.2888271$$

$$Q(p, up) = -4 + 0.9(0.5 * 53.2888271) + 0.9(0.25 * 100) + 0.9(0.25 * 53.2888271)$$
$$= 54.4699583$$

$$Q(p, up) = -4 + 0.9(0.5 * 54.4699583) + 0.9(0.25 * 100) + 0.9(0.25 * 54.4699583)$$
$$= 55.26722192$$

$$Q(p, up) = -4 + 0.9(0.5 * 55.26722192) + 0.9(0.25 * 100) + 0.9(0.25 * 55.26722192)$$
$$= \mathbf{55.80537479}$$

$$Q(p, down) = -4 + 0.9(0.5 * -100) + 0.9(0.25 * 100) + 0.9(0.25 * 0) = \mathbf{-26.5}$$

$$Q(p, down) = -4 + 0.9(0.5 * -100) + 0.9(0.25 * 100) + 0.9(0.25 * -26.5) = -32.4625$$

$$Q(p, down) = -4 + 0.9(0.5 * -100) + 0.9(0.25 * 100) + 0.9(0.25 * -32.4625) = -33.8040625$$

$$Q(p, down) = -4 + 0.9(0.5 * -100) + 0.9(0.25 * 100) + 0.9(0.25 * -33.8040625)$$
$$= -34.10591406$$

$$Q(p, down) = -4 + 0.9(0.5 * -100) + 0.9(0.25 * 100) + 0.9(0.25 * -34.10591406)$$
$$= -34.17383060$$

$$Q(p, down) = -4 + 0.9(0.5 * -100) + 0.9(0.25 * 100) + 0.9(0.25 * -34.17383060)$$
$$= -34.1891119$$

The maximum value of all the possible actions is Up with 55.80537479.

Therefore, going up is the optimal policy. This makes sense because there is a 50% chance that the agent will continue to go up until the 25% that they may go right occurs and they reach the goal. This a risk adverse strategy because if the agent chose to go right, it would be successful only 50% of the time and they have a 25% chance of going down and reaching the monster.

Question 3:

$$R = \begin{matrix} -4 & -4 & -100 & 0 \\ -4 & -4 & 0 & +100 \\ -4 & 0 & -100 & +100 \\ 0 & -4 & -100 & +100 \end{matrix}$$

$$Q = \begin{matrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{matrix}$$

Game 1:

$$Q(1,2) = Q(1,2) + \propto \left(R(1) + \gamma \max[Q(2,4), Q(2,1), Q(2,2)] - Q(1,2)\right)$$
$$= 0 + 0.6(-4 + 0.9(0) - 0)$$
$$= -2.4$$

$$Q(2,4) = Q(2,4) + \propto \left(R(2,4) + \gamma \max[Q(4,2), Q(4,3), Q(4,4)] - Q(2,4)\right)$$
$$= 0 + 0.6(100 + 0.9(0) - 0)$$
$$= 60$$

Game 2:

$$Q(1,2) = Q(1,2) + \propto \left(R(1) + \gamma \max[Q(2,4), Q(2,1), Q(2,2)] - Q(1,2)\right)$$
$$-2.4 + 0.6\left(-4 + 0.9(60) - (-2.4)\right)$$
$$29.04$$

$$Q(2,2) = Q(2,2) + \propto \left(R(2,2) + \gamma \max[Q(2,2), Q(2,1), Q(2,4)] - Q(2,2)\right)$$
$$= 0 + 0.6(-4 + 0.9(60) - 0)$$
$$= 30$$

$$Q(2,1) = Q(2,1) + \propto \left(R(2,1) + \gamma \max[Q(1,2), Q(1,1), Q(1,3)] - Q(2,1)\right)$$
$$= 0 + 0.6(-4 + 0.9(29.04) - 0)$$
$$= 13.2816$$

$$Q(1,3) = Q(1,3) + \propto \left(R(1,3) + \gamma \max[Q(3,1), Q(3,3), Q(3,4)] - Q(1,3)\right)$$
$$= 0 + 0.6(-100 + 0.9(0) - 0)$$
$$= -60$$

Game 3:

$$Q(1,3) = Q(1,3) + \propto \left(R(1,3) + \gamma \max[Q(3,1), Q(3,3), Q(3,4)] - Q(1,3)\right)$$
$$= -60 + 0.6\left(-100 + 0.9(0) - (-60)\right)$$
$$= -84$$

Game 4:

$$Q(1,2) = Q(1,2) + \propto \left(R(1) + \gamma \max[Q(2,4), Q(2,1), Q(2,2)] - Q(1,2)\right)$$
$$= 29.04 + 0.6(-4 + 0.9(60) - 29.04)$$
$$41.616$$

$$Q(2,4) = Q(2,4) + \propto \left(R(2,4) + \gamma \max[Q(4,2), Q(4,3), Q(4,4)] - Q(2,4)\right)$$
$$= 60 + 0.6(100 + 0.9(0) - 60)$$
$$= 84$$

Game 5:

$$Q(1,2) = Q(1,2) + \propto \left(R(1) + \gamma \max[Q(2,4), Q(2,1), Q(2,2)] - Q(1,2)\right)$$
$$= 41.616 + 0.6(-4 + 0.9(84) - 41.616)$$
$$= 59.6064$$

$$Q(2,4) = Q(2,4) + \propto \left(R(2,4) + \gamma \max[Q(4,2), Q(4,3), Q(4,4)] - Q(2,4)\right)$$
$$= 84 + 0.6(100 + 0.9(0) - 84)$$
$$= 93.6$$

$$Q = \begin{matrix} 0 & 59.6064 & -84 & 0 \\ 13.2816 & 30 & 0 & 93.6 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{matrix}$$

Pick the max values of each row

Q[1,2] = 59.6064

Q[2,4] = 93.6

Therefore, the optimal policy is 1 -> 2 -> 4.

Question 4:

I am choosing to do the robot puppy for a household pet.

Environment: The dog would be in a partially observable environment. The dog will be relatively smaller to its environment so it will not be able to see the whole environment. Also, if there are doors the dog is not able to know what is beyond the door unless it has already experienced going through the door.

Actuators: The robot puppy would affect its environment by movement and sounds. Since it is learning its new environment, it would have to learn what is an obstacle is such as furniture, walls, stairs and stairs in order to navigate within the house successfully. To be able to do this, the dog would have legs which would produce movement and "eyes", which may be in the form of a camera to learn its environment. As the robot moves around the environment, it causes change since it will be in a different state each time.

Sensors: It would need to be aware if it is approaching an obstacle, so if it sees a couch it needs to know that it is present to know what state allows it to avoid it. The dog would also need to aware of what type of terrain it is on to adjust how fast it walks. Hardwood is more slippery than carpet so it would have to tread carefully so that it does not fall over. Also, it needs to be able to hear commands from its owner so it knows how to react and also to know if their decisions are acceptable.

Performance Metric: The agent will know that it is making a good decision by learning through reinforcement learning. This resembles training a puppy. If the dog makes an action of "ripping" the furniture so the owner will be able to scold the dog so it knows that it is doing something wrong. If the dog listens to the command of the owner to "sit" then it would be presented with a treat

Algorithms:

- Q-Learning: The dog does not have previous knowledge of its environment so it is able to use the Q values to determine the next step randomly. The dog will be able to remember its previous states because of reinforcement learning will allow it to remember the previous states before and the outcome of them.
- Policy Iteration: The dog would be better off following a policy because it can act upon a change of state (if they hit a wall, they can easily re-calculate a next move) until it reaches its goal. The dog will stop its actions once the policy cannot be improved no longer.

I think these algorithms are suitable for the robot puppy because it encourages learning its environment independently. Since policy iteration has to solve large linear equations, the computation rate may be slower than Q-Learning. If the case of learning faster was a factor, I would choose Q-Learning. But, learning well would be more important. To maximise the performance metric, I would choose policy iteration because in the case the dog gets into a state it cannot recognize, it can create a new policy.