

MDP Report

Purpose:

The purpose of this assignment is to further our understanding of Markov Decision Processes. In particular, this was an opportunity to apply MDP to impact the path that one takes in order to maximize utility. In essence, our goal is to use MDP in order to maximize utility, perhaps ending at a terminal state that yields the greatest utility(i.e hopefully MDP will lead us to the apple). Procedure: The biggest modification to the code was to allow for the horse to jump. This was achieved by adding a variable of boolean type in the T function. If this variable, labeled 'jump' was set to True, the code would enter a condition inside the T function that specifies the corresponding transition probabilities. If 'jump' was set to false, T would function normally. In order to account for the jump action, I also needed to modify the orientations list in the utils.py file. Likewise, there is a boolean variable called jump that if set to true, adds the new set of actions(i.e (0,2,(-2,0)..etc) to the list. To run the required experiments I also added a few lines of code to print the index and its corresponding utility in a grid format. The format represents the grid given in the handout, but because of the index manipulation already done in the file, the grid is upside down. In the experiments, I ran both value and policy iteration for reward values of $\{-0.99, 0, 0.99\}$ and gamma values of $\{0.4, 0.9\}$.

Data:

The data is a matrix of utility values. These values dictate the actions for a given a state. For example(based on no modifications), if the horse starts at (0,0), given a list of actions for that state $\{(1,0), (0,1)\}$, the horse will choose action (0,1) because the utility value for that state is greater than the utility corresponding to (1,0) action($8.7954068181 > 8.17648535992$). Another piece of data, the actions list, is the set of actions the horse can take given its state location. These tuples are merely directions, rather than actual locations in the matrix. For example, if the horse is at state (2,2) and there is wall to the right of the state, the horse's actions given it's location is $\{(0,1), (-1,0), (0,-1)\}$.



Results:

Note: The paths for each experiment was the same for both value and policy iteration, so the following results pertain to both algorithms.

Living Rewards: When changing the living rewards between $(-1, 1)$, I tested values $\{-0.99, 0, 0.99\}$. Compared with the default value of zero, -0.99 and 0.99 both impacted the path taken by the horse. When the reward is set to -0.99 , the path varies because the penalty of being in the snake state is -0.5 , but with the updated reward, being in a blank state is a heavier penalty, thus the horse lands on the snake state in position (1,1). Conversely, when the reward is set to 0.99 , the horse actually makes its way to the apple. When the reward is -0.99 or 0 , the horse never goes past the barn positioned at (1,4). The paths of each reward parameter is shown on the grid below. Living reward = 0 is denoted by $A_1 \dots A_n$, where n is the order of steps. Living reward = -0.99 is denoted by $B_1 \dots B_n$, where n is the order of steps. Living reward = 0.99 is denoted by $C_1 \dots C_n$, where n is the order of steps.

Here is a table of parameters with corresponding keys:

Reward	γ (Gamma)	Jump?	Key
0	0.9	No	A
-0.99	0.9	No	B
0.99	0.9	No	C
0	0.4	No	D
0	0.9	Yes	E

									C ₁₇ 
									C ₁₆
							C ₁₃	C ₁₄	C ₁₅
	A ₆ B ₆ D ₁₀ E ₄ 						C ₁₂		
	A ₅ B ₅ D ₉	D ₈	D ₇				C ₁₁		
A ₃	A ₄ B ₄ E ₃		D ₆			C ₉	C ₁₀		
A ₂ B ₂	B ₃		D ₅			C ₈			
A ₁ B ₁ C ₁ D ₁ E ₁	C ₂ D ₂ E ₂	C ₃ D ₃	C ₄ D ₄	C ₅	C ₆	C ₇			

Note: This matrix only shows the horse's path and key landmarks in the path

Gamma: Since we were only required to pick one gamma value, I chose something towards the middle of the input range, 0.4. The horse still ended up at the barn, but took a different path to get there. The path is shown on the grid and is denoted by D_n

Jumping action and new Transition Probabilities: Interestingly, the jumping action was not able to reach the apple, but unsurprisingly, it did reach the barn in fewer steps. It was also interesting to see that the horse occasionally did not jump. The path is denoted on the grid as E_n