Machine Learning Assignment 2
Team 72

Manish - 2018101073

Daksh Rawat - 2018101087

Task-1

- ***Markov decision process (MDP)*** is a discrete time stochastic control process. It provides a mathematical framework for modeling decision making in situations where outcomes are partly random and partly under the control of a decision maker. MDPs are useful for studying optimization problems solved via dynamic programming and reinforcement learning.
- ***Value Iteration Algorithm*** : Value iteration is a method of computing an optimal Markov Decision Process policy and its value.Value iteration starts at the "end" and then works backward, refining an estimate of either Q* or V*. There is really no end, so it uses an arbitrary end point. Let Vk be the value function assuming there are k stages to go, and let Qk be the Q-function assuming there are k stages to go. These can be defined recursively. Value iteration starts with an arbitrary function V0 and uses the following equations to get the functions for k+1 stages to go from the functions for k stages to go:

$$U_{i+1}(s) = \max_{a \in A(s)} P(s` \mid s, a)\,(R(s` \mid s, a) + \gamma\, U_i(s`))$$
$$P_{i+1}(s) = \max_{a \in A(s)} P(s` \mid s, a)\,(R(s` \mid s, a) + \gamma\, U_{i+1}(s`))$$

Number of iterations for Task-1 = 126

Following are the Observations/Inferences from the final policy obtained from Task-1-:

- When the Stamina of Lero is 0, there is no other option but to recharge, the same is clearly observed in the final policy.
- When the Stamina of Lero is non zero but the number of Arrows are 0, then Lero has 2 options it can either Recharge or Dodge, but the final policy suggests that Lero always prefers to Dodge. This can be attributed to the fact that dodging may increase the number of arrows that Lero has which can be used in the future to attack the enemy and gain final reward.
- When both Stamina and number of Arrows are non zero Lero prefers to Shoot the enemy, this action can be explained by the fact that shooting will decrease the enemy's health and brings him closer to getting the final reward.
- There are some exceptions with the final rule, the optimal action for states (4,1,1) and (4,2,1) is Dodge according to optimal policy which should be Shoot using the above rule. The reason for this can be that the Enemy is far away from losing so it is more advisable to try to get some more arrows and then try to kill the enemy.
- Another exception is for the state (4,3,1), going by the general rule Lero should Shoot but as the enemy is much far from losing and number of arrows is also not low Lero tries to get some Stamina before trying to eliminate the enemy.

Task-2

Number of iterations for Task-2 Part-1 = 100

Following are the Observations/Inferences from the final policy obtained from Part-1-:

- When the Stamina of Lero is 0, there is no other option but to recharge, the same is clearly observed in the final policy.
- When the Stamina of Lero is non zero but the number of Arrows are 0, then Lero has 2 options it can either Recharge or Dodge, but the final policy suggests that Lero always prefers to Dodge. This can be attributed to the fact that dodging may increase the number of arrows that Lero has which can be used in the future to attack the enemy and gain final reward.
- Another exception is for the state (4,3,1), going by the general rule Lero should Shoot but as the enemy is much far from losing and number of arrows is also not low Lero tries to get some Stamina before trying to eliminate the enemy.

The exceptions that we found in Task-1 do not arise in this task because the step cost(reward) for the Shoot action is much higher (less negative) than other actions, so whenever Lero can, he Shoots, so no exceptions arise.

Number of iterations for Task-2 Part-2 = 5

Following are the Observations/Inferences from the final policy obtained from Task-1-:

- The value of gamma is very less compared to part 1 and as we know gamma is the discount factor which represents what preference the model gives to future reward compared to present reward .
- So in this part there are only 5 iterations whereas in part1 there were 100 iterations which is a huge difference. This is because in this part Lero is not considering the future rewards and becomes satisfied by the present reward easily. Therefore without thinking much about the future, the model converges very early.
- $P_{i+1}(s) = \text{argmax}_{a \in A(s)} \; P(s` \mid s, a)\,(R(s` \mid s, a) + \gamma\, U_{i+1}(s`))$
  As per the above formulae it is clear that if γ is low the amount of change in each iteration is much lower, hence the model converges early.

Number of iterations for Task-2 Part-3= 12

As delta is very small which implies more precision is required and hence the number of iterations has increased.
A small observation in the output is that some actions have changed from part-2, the reason for this is due to floating point errors in python. The utility for both the actions is actually same but an error arises at 10-11th place so action changes.