Machine, Data and Learning

Assignment 2

Jashn Arora

2018114006

Akshit Garg

2018113006

Team - **42**

Details of the Algorithm

Value Iteration Algorithm

Value iteration is a method of computing an optimal MDP policy and its value.

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:

Value Iteration Formula
$U_{i+1}(s) = \max_{a \in A(s)} P(s' \mid s, a)\,(R(s' \mid s, a) + \gamma\, U_i(s'))$

Policy Iteration Formula
$P_{i+1}(s) = \operatorname{argmax}_{a \in A(s)} P(s' \mid s, a)\,(R(s' \mid s, a) + \gamma\, U_{i+1}(s'))$

Inference from *Task1:*

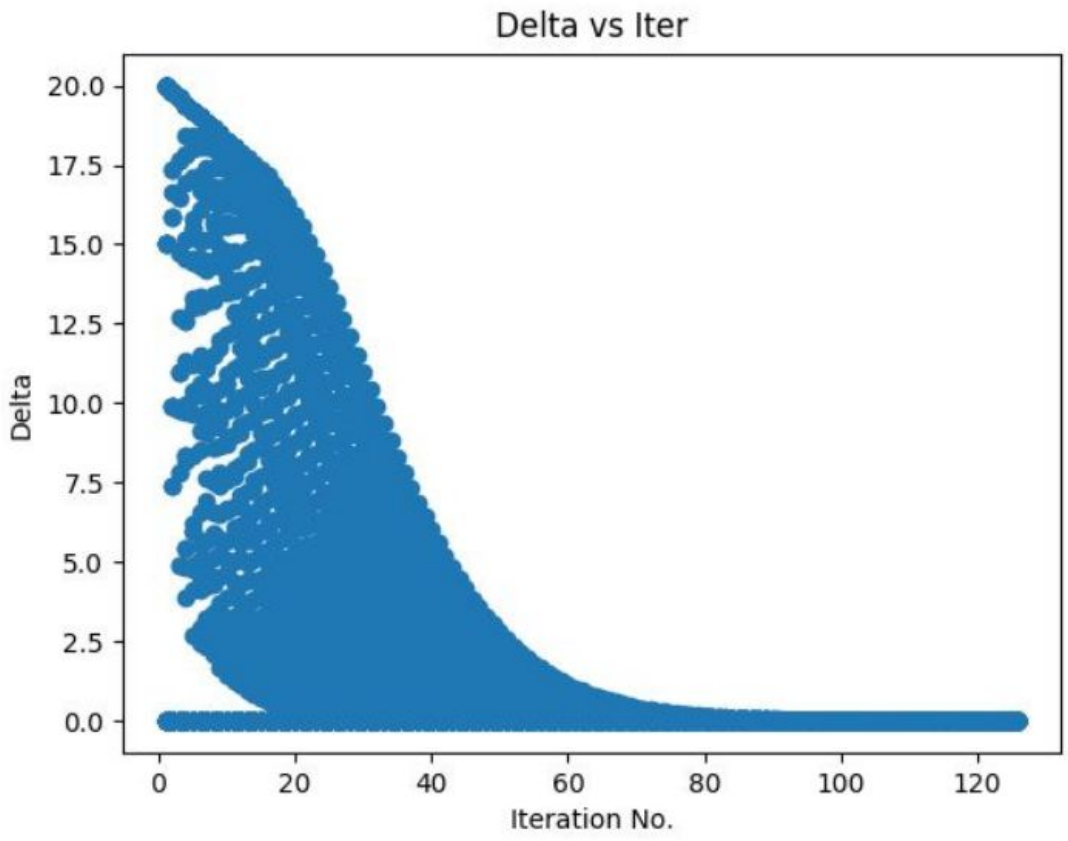
The policy that model follows is fairly obvious.

- Lero has to recharge when his stamino is 0 because that is his only option.
- He tries to shoot when he has stamino and arrows, and when the mighty dragon has less health
- He dodges when he has stamina but needs arrows and the mighty dragon is far from dying
- There are some exceptions like when the state is (4,1,1) according to above rules Lero should shoot but Lero prefers to Dodge according to optimal policy because

enemy is too far away from killing so it is more optimal to gain arrows instead of shooting.

- 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.

If we try to plot Change in Utility (DELTA) vs No. of Iteration plot we can clearly see that delta decreases with increase in number of iterations



Showing that the Algorithm moves towards the convergence as the no. of iterations increase

Inference from *Task2-part1*:

As we decreased the step cost of shooting, we see that the player is **preferring to shoot** whenever he has arrows and has some stamina.

This is due to the low step cost of shooting, hence there is more incentive for the agent to "SHOOT" whenever possible.

Inference from *Task2-part2*:

As we decrease gamma, the player is trying to complete the game as soon as possible. We can see this by the **large decrease in number of iterations.** Now, we only have 5 iterations.

Small Gamma Value encourages the player to give more weightage to current states rather than the future states making him do all the work in current time. So whenever he has arrows, he shoots without thinking of the future, hence reducing the number of iterations it took to converge.

This Lack of future planning leads to Lero's Weird Behaviour

For example, in the state tuple of (3, 3, 1) and (3, 2, 1) [here, (h, a, s) represents the tuple of health, arrows, stamina], the agent changes its policy from "RECHARGE" to "SHOOT". Similarly for the state (4, 2, 1), the agent changes its policy from "DODGE" to "SHOOT". The agent doesn't think whether he will have enough stamina or enough arrows in future to kill the dragon. He just tries to get as much closer to any one of the terminal states as possible from the current state

Inference from *Task2-part3*:

Here, we observe that all the values are same with task-2 part-2 till the 5th iteration because there is no change in gamma or the step cost.

But we see a **growth in the number of iterations** because we stop the algorithm when all the utilities differ by less than Delta now as the delta is reduced the difference b/w all the utilities must be decreased which will happen only when more iterations are performed hence leading to increase in the number of iterations.