COL 778 A2 Report

Kartik Krishna

March 1st 2024

1 Part A

1.1 Value Iteration

I implemented vannila value iteration as discussed in class, with $\epsilon = 0.001$ yielding the optimal policy and values for the smaller map and $\epsilon = 1 \times 10^{-9}$ for the large map.

Figure 1 shows the heat map and the corresponding optimal policy for the small map for vannila value iteration. Figure 2 shows the heat map and the corresponding optimal policy for the large map for vannila value iteration

1.2 Asynchronous Prioritized Sweep

The algorithm for prioritized sweep is as shown in 3.Here, a queue is maintained of every state-action pair whose estimated value would change non-trivially if backed up, prioritized by the size of the change. When the top pair in the queue is backed up, the effect on each of its predecessor pairs is computed. If the effect is greater than some small threshold, then the pair is inserted in the queue with the new priority (if there is a previous entry of the pair in the queue, then insertion results in only the higher priority entry's remaining in the queue). In this way the effects of changes are efficiently propagated backward until quiescence

The convergence criterion for prioritized sweeping too is defined by thresholding max norm of two values after two iterations which means that convergence criteria is evaluated only after all the states have been updated. In case of small map with $\epsilon = 0.001$ yields the optimal policy and values and $\epsilon = 1 \times 10^{-9}$ yields the optimal policy and values for the large map.

Figure 4 and Figure 5 show the heat and the corresponding optimal directions for each state for small and large map

1.3 Comparision of Vannila vs Prioritized sweep

As we can see from Table 1 and Table 2, Prioritized sweep takes much more time to converge to the optimal policy than the vannila value iteration because updates of states in prioritized sweep is prioritized based on bellman error and

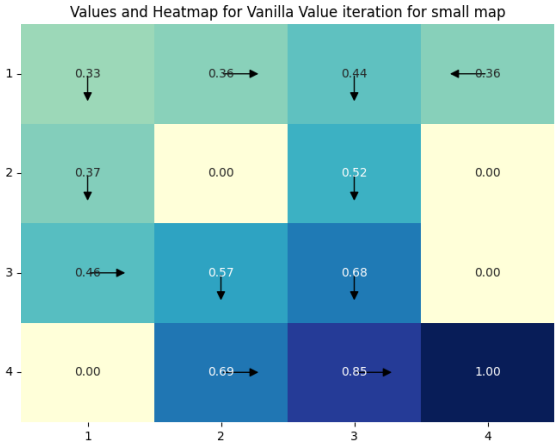

Figure 1: Vannila Value iteration for small map heat map and directions

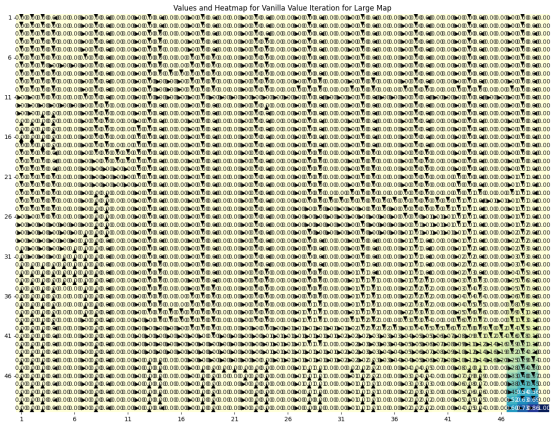

Figure 2: Vannila Value iteration for large map heat map and directions

Table 1: Comparison of Algorithms for small map

small map	Number of iterations	Updates	Time Taken (in sec)
Vannila Value Iteration	12	132	0.006
Prioritized Sweep	14	144	0.006

Initialize $Q(s,a)$, $Model(s,a)$, for all s,a, and $PQueue$ to empty
Do forever:
(a) $s \leftarrow$ current (nonterminal) state
(b) $a \leftarrow policy(s,Q)$
(c) Execute action a; observe resultant state, s', and reward, r
(d) $Model(s,a) \leftarrow s', r$
(e) $p \leftarrow |r + \gamma \max_{a'} Q(s',a') - Q(s,a)|$.
(f) if $p > \theta$, then insert s,a into $PQueue$ with priority p
(g) Repeat N times, while $PQueue$ is not empty:
$s,a \leftarrow first(PQueue)$
$s', r \leftarrow Model(s,a)$
$Q(s,a) \leftarrow Q(s,a) + \alpha\big[r + \gamma \max_{a'} Q(s',a') - Q(s,a)\big]$
Repeat, for all $\bar{s},\bar{a}$ predicted to lead to s:
$\bar{r} \leftarrow$ predicted reward
$p \leftarrow |\bar{r} + \gamma \max_{a} Q(s,a) - Q(\bar{s},\bar{a})|$.
if $p > \theta$ then insert $\bar{s},\bar{a}$ into $PQueue$ with priority p

Figure 3: Algorithm for prioritized sweep

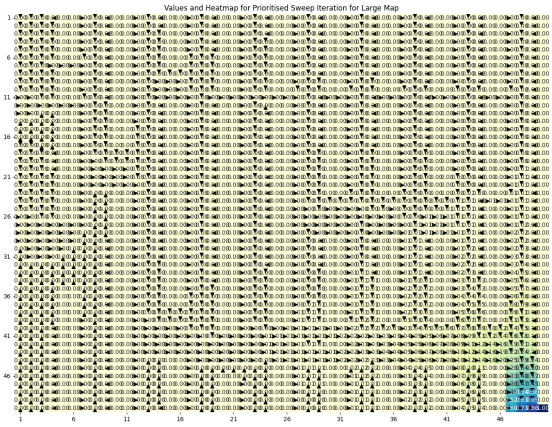

Figure 4: Values and Heatmap for Prioritised Sweep Iteration Large Map

Table 2: Comparison of Algorithms for large map

large map	Number of iterations	Updates	Time Taken (in sec)
Vannila Value Iteration	128	217856	4.84
Prioritized Sweep	229	388064	7.83

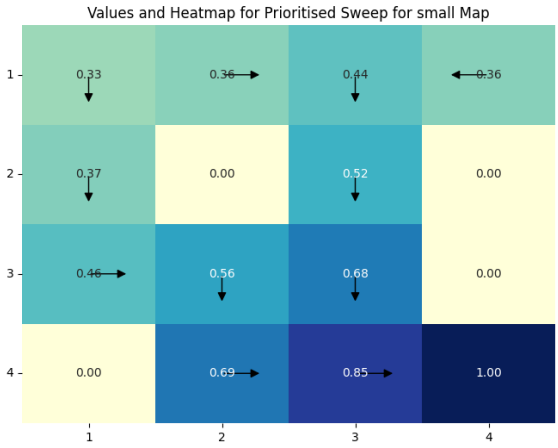

Figure 5: Values and Heatmap for Prioritised Sweep Iteration small Map

hence some states (for example start state) take more number of updates to converge to the optimal values. And hence with larger grids the amount of time taken and the number of updates required to reach the optimal policy is high in prioritized sweep.

1.4 Policy Iteration

Figure 7 and Figure 6 show the heat map for optimal policy directions for each state for policy iteration For policy iteration, in case of the small map, the time taken for convergence was 0.003 seconds and the total number of iterations needed for reaching to the optimal policy was 3 iterations with $\epsilon = 0.001$.

For the large map, the time taken for convergence was 7.43 seconds and the number of iterations required for converging to the optimal policy was 12 steps with $\epsilon = 1 \times 10^{-9}$

While the number of iterations required for converging to the optimal policy for policy iteration is less than the number of iterations required for converging in case of value iteration, the time taken for converging remains the same because in policy iteration, during the policy evaluation stage for a given policy we iterate through the states and find the best values until convergence and during the improvement stage for given values of states we try to find the best policy. Theoretically, policy iteration should converge faster because we reach the optimal policy before convergence of values.

Hence if the number of actions are less for larger grids, policy iteration is better and if the number of actions is more then value iteration is better.

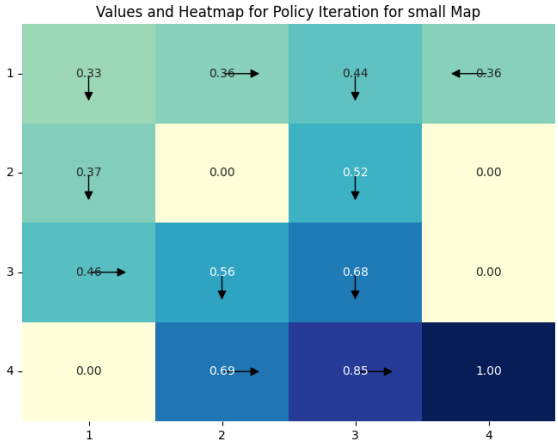

Figure 6: Values and Heatmap for Policy Iteration Iteration small Map

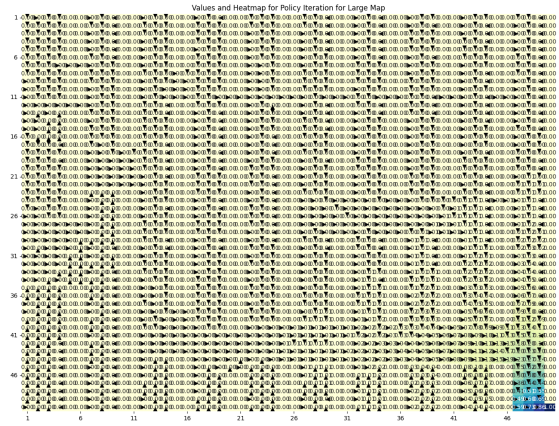

Figure 7: Values and Heatmap for Policy Iteration Iteration large Map

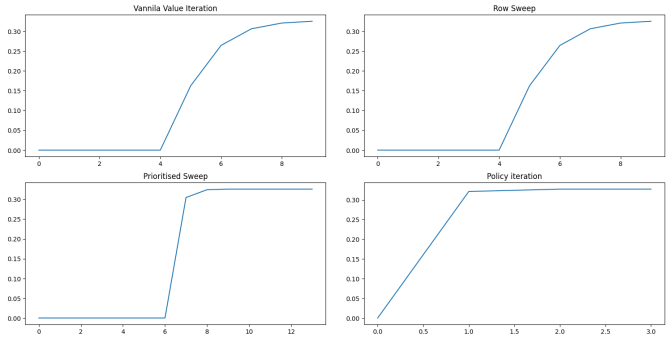

Figure 8: Value of start state at different iterations for small map

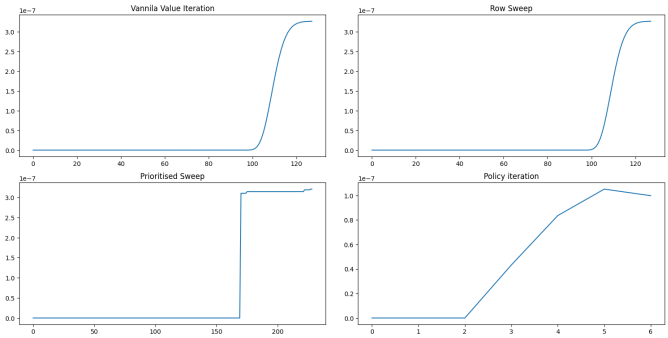

Figure 9: Value of start state at different iterations for large map

1.5 Start State values at different iterations

Figure 8 and Figure 9 show the plots for start state values at different iterations for large map and small map for all the algorithms

2 Part B

2.1 Living Reward Analysis

When we have negative living reward of -0.1, the value iteration converges to the optimal policy i.e, the agent reaches the goal state.The heat map and directions after convergence to optimal policy is shown in Figure 10. When the living reward is made -0.9, agent doesn't reach the goal state but instead reaches the nearest hole because the reward obtained at hole is 0.

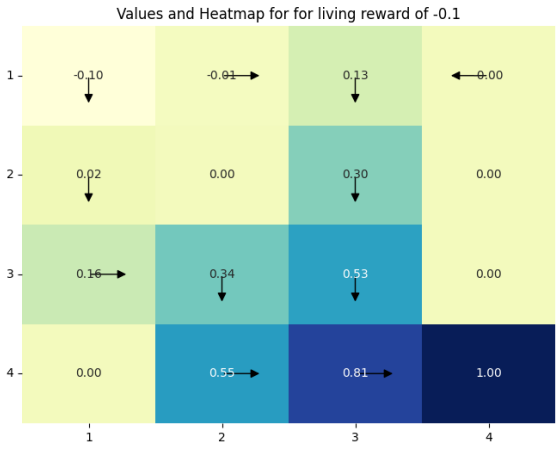

Figure 10: Heat map and directions for living reward of -0.1

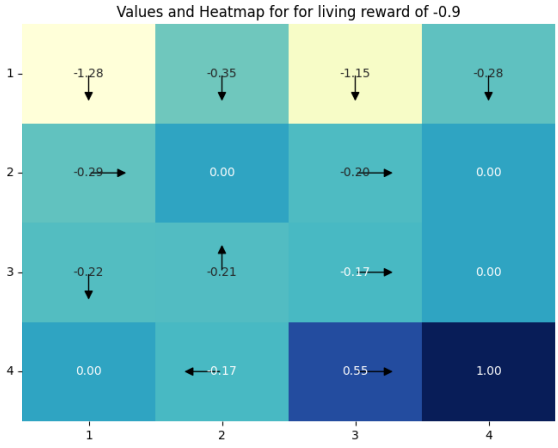

Figure 11: Heat map and directions for living reward of -0.9

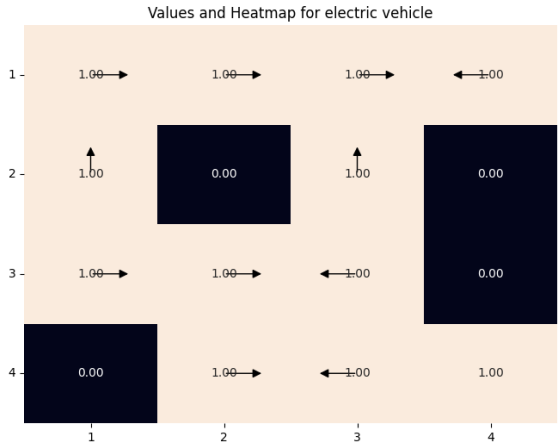

Figure 12: Electric vehicle heat map and directions

2.2 Analysis for the Electric Vehicle

For the electric vehicle, when the living reward is 0.001 and the discount factor is 0.999, the value of all the states after reaching convergence is close to 1 and hence the vehicle never reaches the goal state and the policies at all the states are random since all values are close to 1 and all states are optimal states.

Figure 12 shows the heat map and the directions for each state

2.3 Changing transition probabilities

Figure 13 shows the heat map and intended direction of states when the road is slippery. Here we can see that since the road is slippery due to high uncertainty in the transition model, the car goes in the opposite direction to the actual optimal direction and hence would reach the goal in a probabilistic manner.

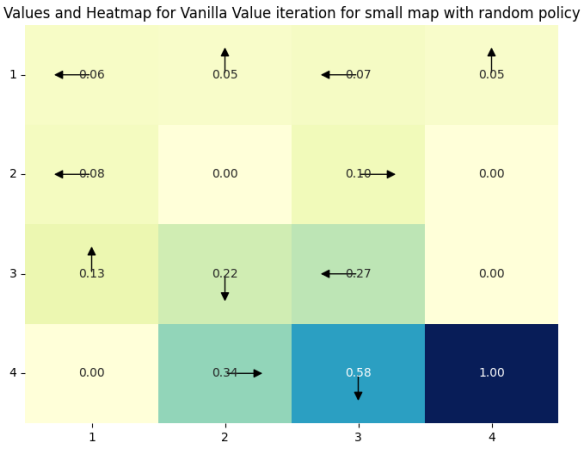

Figure 13: Values and Heatmap for Vanilla Value iteration for small map with stochastic policy