Artificial Intelligence

Assignment 2 (handwritten)

Part B

(TEAM 51)

RAMKISHORE SARAVANAN
20161092
AAMIR FARHAN
20161078

Different policies acquired by the Value Iteration algorithm and their corresponding Utilities

`* > < <`
`v < * <`
`< < < v`
`^ ^ > *`
	`-51.000 -10.200 -10.200 -10.200`
	`-10.200 -10.200 0.000 -10.200`
	`-5.100 -10.200 -10.200 -10.200`
	`-10.200 -15.300 -10.200 51.000`

`* > < <`
`v < * v`
`< < v v`
`^ > > *`
	`-51.000 -20.400 -20.400 -20.400`
	`-16.320 -20.400 0.000 -20.400`
	`-11.220 -16.830 -20.400 28.560`
	`-16.830 -25.500 28.560 51.000`

`* v < v`
`v < * v`
`< > > v`
`> > > *`
	`-51.000 -30.600 -30.600 -30.600`
	`-22.848 -26.979 0.000 8.568`
	`-17.391 -23.766 13.821 31.416`
	`-23.409 3.315 31.416 51.000`

`* v > v`
`v v * v`
`> > > v`
`> > > *`
	`-51.000 -39.943 -40.800 -9.466`
	`-29.095 -33.915 0.000 16.646`
	`-23.639 -1.510 19.457 35.124`
	`-11.628 7.788 35.124 51.000`

`* > > v`
`v v * v`
`> > > v`
`> > > *`
	`-51.000 -27.639 -10.384 6.822`
	`-19.664 -7.467 0.000 23.612`
	`-1.160 9.656 25.146 36.713`
	`1.088 16.255 36.713 51.000`

`Final`	`-51.000 -14.210 -3.010 9.740`
	`-8.071 -1.809 0.000 24.083`
	`3.897 11.723 25.500 36.833`
	`4.249 17.043 36.833 51.000`

The algorithm took 57 iterations to converge changing over 5 different policies.

Observations of policy changes

Initial Policy

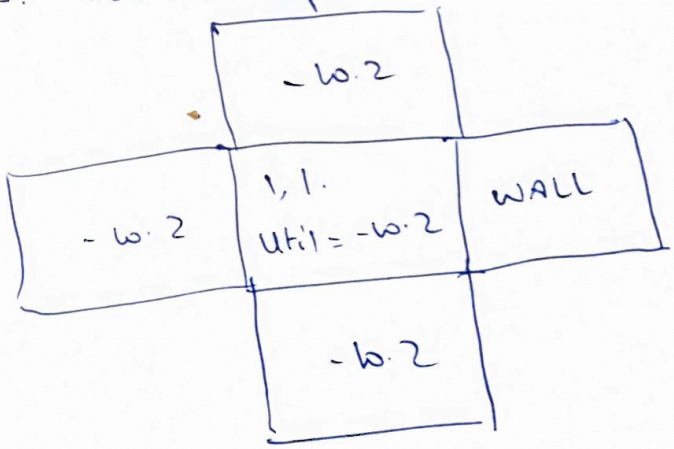
Policy:

```
* > < <
v < * <
< < < v
^ ^ > *
```

Corresponding Utilities:

```
-51.000 -10.200 -10.200 -10.200
-10.200 -10.200 0.000 -10.200
-5.100 -10.200 -10.200 -10.200
-10.200 -15.300 -10.200 51.000
```

- Most of the states have a utility of -0.2 which is equal to the unit step reward.
- Terminal states have their R(s) values as utilities while non terminal states with R(s) values have utility(s) = R(s) + step reward.
- The policy is indifferent to actions in many states. For example consider state 1,1.



irrespective of the action it takes, its gain will be the same.

- Also, note how the policy chooses indefinitely looping b/w (0,1) and (0,2) and not reaching (0,0) which have the utility of -51.

2nd Policy:

Policy:

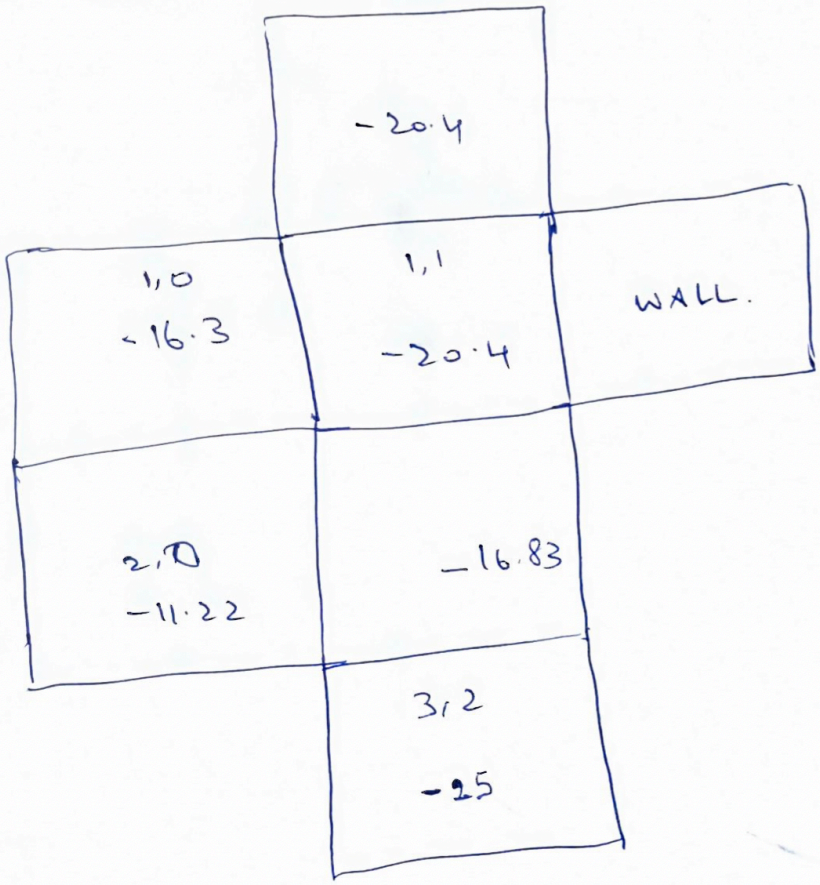
```
* > < <
v < * v
< < v v
^ > > *
```

Corresponding Utilities:

```
-51.000 -20.400 -20.400 -20.400
-16.320 -20.400 0.000 -20.400
-11.220 -16.830 -20.400 28.560
-16.830 -25.500 28.560 51.000
```

- Most states have -20.4 as their utility value, which is twice the step reward.
- Rewards have propogated by 1 level. i.e., states in initial policy whose R(s) ≠ step reward have affected the adjacent states.

again, lets consider state 1,1.



Now, the agent can make a clear decision to move to left because of the immediate reward in (2,0) affected utility of state (1,0)

- However, moves are more greedy than optimal, since the world penalises highly for each step (step reward = -10.2), moving to terminal state (3,3) should be the optimal move.

- Similarly, moves in (1,3), (2,2), (3,1) have changed, all of them affected by the higher utility of terminal state (3,3).

~~It can be noted that~~

3rd Policy

Policy:

```
* v > v
v v * v
> > > v
> > > *
```

Corresponding Utilities:

```
-51.000 -30.600 -30.600 -30.600
-22.848 -26.979 0.000 8.568
-17.391 -23.766 13.821 31.416
-23.409 3.315 31.416 51.000
```

- Now most states have decided that optimal move should be in the direction of terminal state, and very few states, ~~(5~~ (3 as compared to 6 in 2nd & 4 in 1st) are unaffected by rewards of other states, and have utility = 3 times step reward.

- The indefinite loop (0,1) & (0,2) is broken, as the reward of state (2,0) affected (0,1) to move down than avoid (0,0) blindly.

- Even more states now point to terminal state (3,0).

- State (2,0) tries to move left to maximize its probability of staying in the same state, since all adjacent states have lesser utility, effect of terminal state has not reached it yet.

- State (2,1) now points right rather than left, choosing closeness to (3,3) more than the greedy option (2,0). The same goes also for (3,0), even though the immediate block has R(s) = -5.1.

- Also, (2,2) changed from pointing down to pointing right. This can be explained because of the proximity of (3,2) to state (3,1) with R(s) = -5.1.

- State (1,3) is also pointing to terminal state now.

- The R(s) values have affected upto 4 block radius and the policy is more optimal relatively.

4th Policy.

Policy:

```
* > < <
v < * v
< < v v
^ > > *
```

Corresponding Utilities:

```
-51.000 -39.943 -40.800 -9.466
-29.095 -33.915 0.000 16.646
-23.639 -1.510 19.457 35.124
-11.628 7.788 35.124 51.000
```

- Only one state remains unaffected by R(s) values of other states, (0,2) with utility = 4 times step_reward.
- The indefinite strategy of minimizing loss in state (2,2) is broken, the algorithm decided that incurring immediate loss also leads to chances of high reward at terminal state (3,3).
- State (1,1) is also pointing down instead of left, again choosing proximity to terminal state over immediate proximity to lesser reward.
- Move in state (0,2) has become more optimal, because its adjacent states have been affected by reward of state (3,3).
- The only state unaffected by terminal state (3,3) is (0,1), which is 6 steps away.

5th & the final policy:

Policy:

```
* > > v
v v * v
> > > v
> > > *
```

Corresponding Utilities:

```
-51.000 -14.210 -3.010 9.740
-8.071 -1.809 0.000 24.083
3.897 11.723 25.500 36.833
4.249 17.043 36.833 51.000
```

- The only notable change b/w 4th & 5th policy is that the optimal move in state (0,1) changed from down to right.

- Previously it was affected only by lesser penalty ($R(s) = 5 \cdot 1$) of state (2,0) but now, terminal state has proved its dominance.

- All states point towards the state (3,3) as the reward is considerably high.

Actions from start state (3,0): (descriptive version of policies)

⇒ 1st policy:

- try to move up to state (2,0) since it has lesser penalty of -5.1 compared all other adjacent states.

⇒ 2nd policy:

- initially move up, & try to stay in (2,0) permanently.

⇒ from 3rd policy onwards:

- reach the terminal state, with some higher preference to local rewards as the n ~~increases~~ decreases in nth policy.

Grid world for x = 51

-51			
		wall	
5.1			
START	**-5.1**		**51**