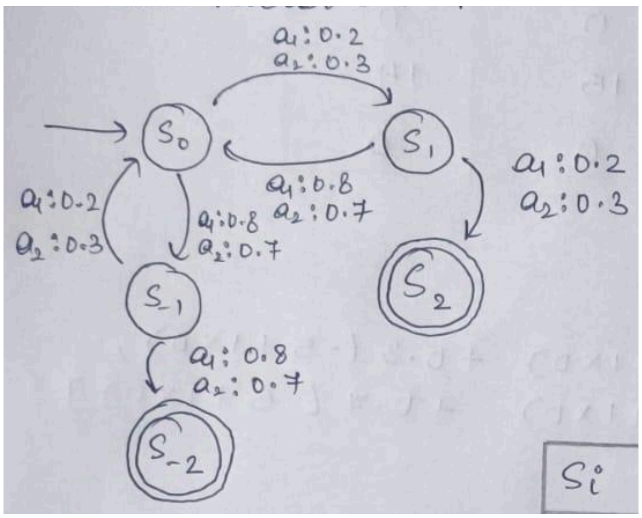
AI Lab Project-1 Report

Abhinav Singh 22UCS004
Akhil Murarka 22UCS009
Ishita Singhal 22UCS090

Introduction:

This report explores a state-based game in which a player navigates through discrete states with rewards based on the actions taken and the final state reached. The game ends upon reaching a terminal state, either -2 or 2. The objective is to use value iteration to determine the _optimal value function $V_{opt}(s)$ for each state s and the corresponding optimal policy π_{opt} after two iterations.

System Model:



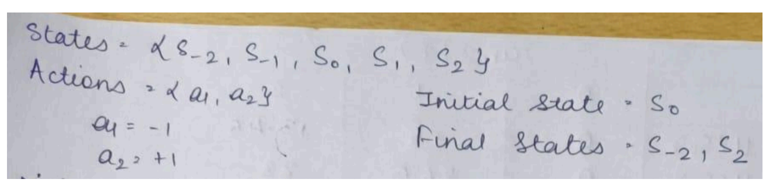
Assuming {-2, -1, 0, +1, +2} as {S_{-2}, S_{-1}, S_0, S_1, S_2} and actions {-1, +1} as {a_1, a_2} respectively. Initial state is S_0 . Final/Terminal states are S_{-2}, S_2.

Action -1(a_1):

- 80% chance of moving to state s_{-1}
- 20% chance of moving to state s_{+1}

Action +1(a_2):

- 70% chance of moving to state s_{-1}
- 30% chance of moving to state s_{+1}



The rewards are:

- Reaching state -2 yields a reward of 20.
- Reaching state 2 yields a reward of 100.
- Any transition to a non-terminal state yields a reward of -5.

Discount factor(γ) is 1.

Implementation:

- **Environment**
  - States: {S_{-2}, S_{-1}, S_0, S_1, S_2}
  - Actions: {a_1, a_2}
  - Transition Function:

S_i	a	S_j	$P(S_j \mid S_i, a)$
S_0	a_1	S_1	0.2
S_0	a_1	S_{-1}	0.8
S_0	a_2	S_1	0.3
S_0	a_2	S_{-1}	0.7
S_1	a_1	S_2	0.2
S_1	a_1	S_0	0.8
S_1	a_2	S_2	0.3
S_1	a_2	S_0	0.7
S_{-1}	a_1	S_0	0.2
S_{-1}	a_1	S_{-2}	0.8
S_{-1}	a_2	S_0	0.3
S_{-1}	a_2	S_{-2}	0.7

  - Reward Function:

Reward function

S	R(s, a, s')
-2	20
-1	-5
0	-5
1	-5
2	100

States = $\{S_{-2}, S_{-1}, S_0, S_1, S_2\}$

Actions = $\{a_1, a_2\}$

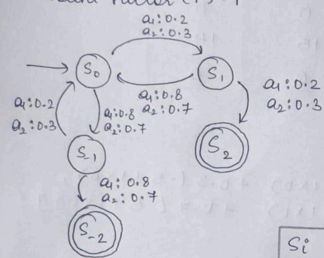
$a_1 = -1$

$a_2 = +1$

Initial state = S_0

Final states = S_{-2}, S_2

Discount Factor (γ) = 1



Reward function

S	R(S, a, S')
-2	20
-1	-5
0	-5
1	-5
2	100

Bellman eq^n:

$$U(S) = \max_a \sum_{S'} P(S'|S,a) \cdot [R(S,a,S') + \gamma U(S')]$$

S_i	a	S_j	$P(S_j \mid S_i, a)$
S_0	a_1	S_1	0.2
S_0	a_1	S_{-1}	0.8
S_0	a_2	S_1	0.3
S_0	a_2	S_{-1}	0.7
S_1	a_1	S_2	0.2
S_1	a_1	S_0	0.8
S_1	a_2	S_2	0.3
S_1	a_2	S_0	0.7
S_{-1}	a_1	S_0	0.2
S_{-1}	a_1	S_{-2}	0.8
S_{-1}	a_2	S_0	0.3
S_{-1}	a_2	S_{-2}	0.7

a)

	time (t)		
	0	1	2
$U(S_0)$	0	-5	13.45
$U(S_1)$	0	26.5	23
$U(S_2)$	0	0	0
$U(S_{-1})$	0	15	14
$U(S_{-2})$	0	0	0

for t = 1

$S = S_0$

$U(S_0) = \max[0.2(-5+1\times0) + 0.8(-5+1\times0), 0.3(-5+1\times0) + 0.7(-5+1\times0)]$

$= \max[-5, -5]$

$= -5$

$S = S_1$

$U(S_1) = \max[0.2(100+1\times0) + 0.8(-5+1\times0), 0.3(100+1\times0) + 0.7(-5+1\times0)]$

$= \max[16, 26.5]$

$= 26.5$

$U(S_2) = 0$

$U(S_{-2}) = 0$ → Terminal states

$S = S_{-1}$

$U(S_{-1}) = \max[0.2(-5+1\times0) + 0.8(20+1\times0), 0.3(-5+1\times0) + 0.7(20+1\times0)]$

$= \max[15, 12.5]$

$= 15$

for t = 2

$S = S_0$

$U(S_0) = \max[0.2(-5+1\times26.5) + 0.8(-5+1\times15), 0.3(-5+1\times26.5) + 0.7(-5+1\times15)]$

$= \max[12.3, 13.45]$

$= 13.45$

$S = S_1$

$U(S_1) = \max[0.2(100+1\times0) + 0.8(-5+1\times-5), 0.3(100+1\times0) + 0.7(-5+1\times-5)]$

$= \max[12, 23]$

$= 23$

$S = S_{-1}$

$U(S_{-1}) = \max[0.2(-5\times1+-5) + 0.8(20+1\times0), 0.3(-5+1\times-5) + 0.7(20+1\times0)]$

$= \max[14, 11]$

$= 14$

$S = S_2$

$U(S_2) = 0$

$S = S_{-2}$

$U(S_{-2}) = 0$

→ Terminal States

- **Output**

Below table represents the utility achieved for each state after 0th, 1st, 2nd iteration. As S_{-2}, S_2 are terminal states so their utility remains zero and for the rest of the states it is calculated using the bellman equation.

	time (t)		
	0	1	2
$U(S_0)$	0	-5	13.45
$U(S_1)$	0	26.5	23
$U(S_2)$	0	0	0
$U(S_{-1})$	0	15	14
$U(S_{-2})$	0	0	0

Below table shows the Optimal policies for the non terminal states

b)

S	$\pi^*(S)$
S_0	a_2
S_{-1}	a_1
S_1	a_2

Conclusion: Through this value iteration process, we can establish both the optimal state values and the corresponding policy for decision-making in this state-based game. This iterative process demonstrates how MDP-based methods like value iteration can efficiently determine optimal strategies in games with probabilistic transitions and rewards.

[Code Repository](#)