Value Iteration:

Finding optimal policies of MDPs when we have transition probabilities and Reward functions

Bellman Equation gives us a recursive definition of the optimal value

$$V^*(s) = \max_{a \in A} \sum_{s' \in S} P(s' \mid s, a) \left[R(s, a, s') + \gamma V^*(s') \right]$$

Dynamic Programming

In value iteration convert this equation into an update rule (recursive def.)

→ Iterative process → Refine $V^*(s)$ → Iterate Bellman update until convergence

→ start with $V_0(s) = 0$ for all states s

→ Iterate Bellman update until convergence:

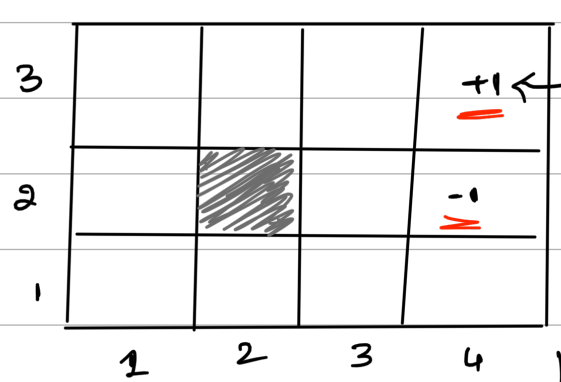
$$V_{i+1}(s) \leftarrow \max_{a \in A} \sum_{s' \in S} P(s' \mid s, a) \left[R(s, a, s') + \gamma V_i(s') \right]$$

$i+1$: iteration number

Bellman Update Rule

as $i \to \infty$ we get $V^*(s)$

Example MDP



reward given in terminal state

living reward: every time step in env is not penalized or given a reward

$\gamma = 0.9$

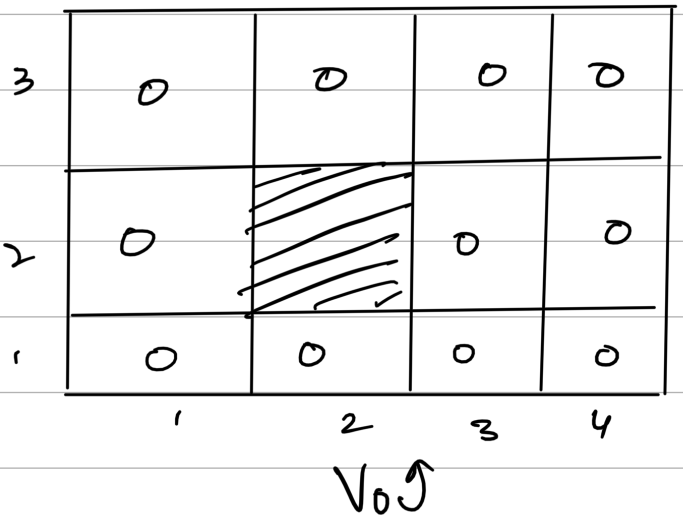
living reward = 0

transition where 80% correct action is executed ← noise = 0.2

probability

0^{th} iteration :-

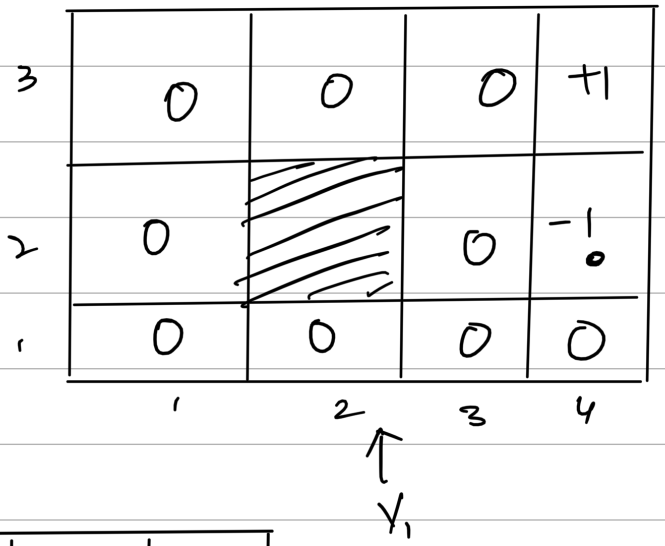
① Initializing the estimate of optimal value function start with $V_0(S) = 0$



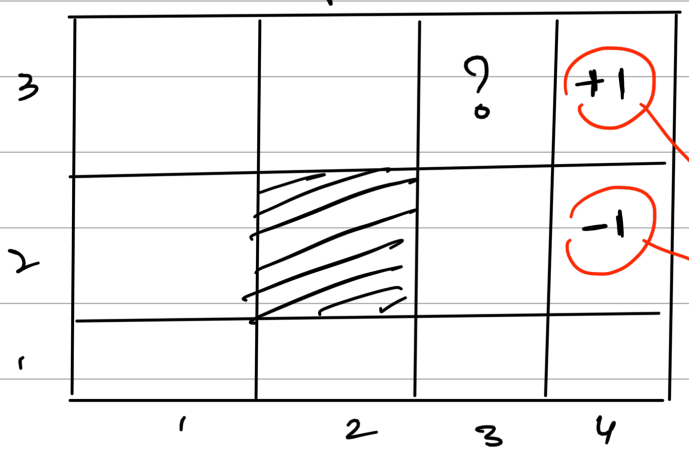
V_0↑

Plug V_0 into Bellman update Rule, we get

$$V_{i+1}(s) \leftarrow \max_{a \in A} \sum_{s' \in S} P(s'|s,a)[R(s,a,s') + \gamma V_i(s')]$$



V_2 ↓ V_1 ↑



$V_2(\langle 3,4 \rangle) \leftarrow 1$
$V_2(\langle 2,4 \rangle) \leftarrow -1$
} terminal states

no new rewards since $\langle 3,4 \rangle$ & $\langle 2,4 \rangle$ are terminal states so set $V_2(\langle 3,4 \rangle)$ & $V_2(\langle 2,4 \rangle)$

to +1, -1 resp.

$$V_2(\langle 3,3\rangle) \leftarrow \sum_{s' \in S} P(s' \mid \langle 3,3\rangle, \text{right}) \left[r(\langle 3,3\rangle, \text{right}, s') + 0.9\, V_1(s') \right]$$

$$\leftarrow 0.8[0 + 0.9 \times 1] + 0.1(0 + 0.9 \times 0) + 0.1(0 + 0.9 \times 0)$$

$$= 0.72.$$

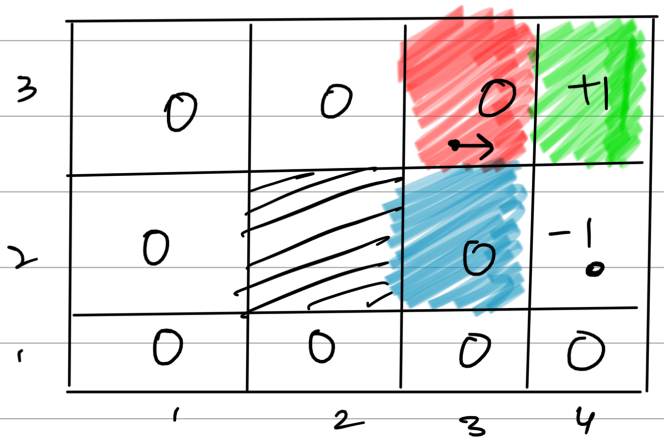
At $\langle 3,3\rangle$ move towards right –
end up at $\langle 3,4\rangle$ or $\langle 2,3\rangle$

At $\langle 3,3\rangle$ move towards left:
end up at $\langle 3,2\rangle$ or $\langle 2,3\rangle$

At $\langle 3,3\rangle$ move up
end up at $\langle 3,3\rangle$

Consider moving right from $\langle 3,3\rangle$

optimal action = 'right'



0	0	0.72	+1
0	X	0	-1
0	0	0	0

V_2

V_3

0	0.52	0.78	+1
0	X	0.43	-1
0	0	0	0

* Information propagates outward from terminal states
* Eventually all states have correct value estimates

Bellman Eq is a property of optimal value function
Use Bellman function as an update rule and we end up with optimal value function

Value Iteration converges to optimal value function

Pseudocode of value iteration

① start with $V_0(s) = 0$
② Iterate until convergence:

$$V_{i+1}(s) \leftarrow \max_{a \in A} \sum_{s' \in S} P(s'|s,a)\left[R(s,a,s') + \gamma V_i(s')\right]$$

$$V_3(s) \leftarrow \max_{a \in A} \sum_{s' \in S} P(s'|s,a)\left[R(s,a,s') + \gamma V_2(s')\right]$$

Value Iteration:

↳ In value iteration, because of the max operation, the equations are not linear anymore.
↳ Use an iterative procedure to solve them.
↳ Initialize the utility of every state = 0 ($\gamma = 0.5$)
↳ loop through states using Bellman equation

$$V(S) = r(S) + \gamma \max_a \left[\sum_{s'} P(s'|s,a) V(s') \right]$$

when $s = 0$

$$V(0) = r(0) + \gamma \max_a \left[\sum_{s'} P(s'|s=0, a) V(s') \right]$$

$$= r(0) + \gamma \max \begin{bmatrix} \sum_{s'} P(s'|s=0, a=\text{up}) V(s'), \\ \sum_{s'} P(s'|s=0, a=\text{left}) V(s'), \\ \sum_{s'} P(s'|s=0, a=\text{down}) V(s'), \\ \sum_{s'} P(s=0, a=\text{right}) V(s') \end{bmatrix}$$

$$= r(0) + \gamma \max \begin{bmatrix} 0.8\,V(0) + 0.1\,V(0) + 0.1\,V(1) \\ 0.8\,V(0) + 0.1\,V(4) + 0.1\,V(0) \\ 0.8\,V(4) + 0.1\,V(1) + 0.1\,V(0) \\ 0.8\,V(1) + 0.1\,V(0) + 0.1\,V(4) \end{bmatrix}$$

$$= -0.04 + 0.5 \times \max \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = -0.04$$

in-place procedure, when we see $V(0)$ then $V(0) = -0.04$ instead of 0 [prev value of $V(0)$]

Now when $s = 1$,

$$V(1) = r(1) + \gamma \max \begin{bmatrix} 0.8\,V(1) + 0.1\,V(0) + 0.1\,V(2) \\ 0.8\,V(0) + 0.1\,V(1) + 0.1\,V(1) \\ 0.8\,V(1) + 0.1\,V(2) + 0.1\,V(0) \\ 0.8\,V(2) + 0.1\,V(1) + 0.1\,V(1) \end{bmatrix}$$

$$= -0.04 + 0.5 \times \max \begin{bmatrix} 0.1 \times (-0.04) \\ 0.8 \times (-0.04) \\ 0.1 \times (-0.04) \\ 0 \end{bmatrix} = -0.04$$

Repeat for states 2 – 11, we get this utility

S = 0 V = -0.04	S = 1 V = -0.04	S = 2 V = -0.04	S = 3 V = 1.0
S = 4 V = -0.04		S = 6 V = -0.04	S = 7 V = -1.0
S = 8 V = -0.04	S = 9 V = -0.04	S = 10 V = -.042	S = 11 V = -.0421

↑ utility value after the first iteration

utility value after 11 iterations

S = 0 V = 0.0897	S = 1 V = 0.3147	S = 2 V = 0.8093	S = 3 V = 1.999
S = 4 V = -0.0046		S = 6 V = 0.1935	S = 7 V = -1.9990
S = 8 V = -0.0456	S = 9 V = -0.0301	S = 10 V = 0.0324	S = 11 V = -0.0698

Get optimal policy :-

$$\pi(s) \leftarrow \underset{a}{\operatorname{argmax}} \left[\sum_{s'} p(s' \mid s, a)\, v(s') \right]$$

get optimal policy by applying ↑ equation for each state