



# Reinforcement Learning

## Computer Engineering Department

## Sharif University of Technology

**Mohammad Hossein Rohban, Ph.D.**

Spring 2024

Courtesy: Some slides are adopted from CS 285 Berkeley, and CS 234 Stanford, and Pieter Abbeel's compact series on RL.



# Value Iteration

- $V_0^*(s)$  = optimal value for state  $s$  when  $H=0$ 
  - $V_0^*(s) = 0 \quad \forall s$
- $V_1^*(s)$  = optimal value for state  $s$  when  $H=1$ 
  - $V_1^*(s) = \max_a \sum_{s'} P(s'|s, a)(R(s, a, s') + \gamma V_0^*(s'))$
- $V_2^*(s)$  = optimal value for state  $s$  when  $H=2$ 
  - $V_2^*(s) = \max_a \sum_{s'} P(s'|s, a)(R(s, a, s') + \gamma V_1^*(s'))$
- $V_k^*(s)$  = optimal value for state  $s$  when  $H = k$ 
  - $V_k^*(s) = \max_a \sum_{s'} P(s'|s, a)(R(s, a, s') + \gamma V_{k-1}^*(s'))$

# Value Iteration

Algorithm:

Start with  $V_0^*(s) = 0$  for all  $s$ .

For  $k = 1, \dots, H$ :

For all states  $s$  in  $S$ :

$$V_k^*(s) \leftarrow \max_a \sum_{s'} P(s'|s, a) (R(s, a, s') + \gamma V_{k-1}^*(s'))$$

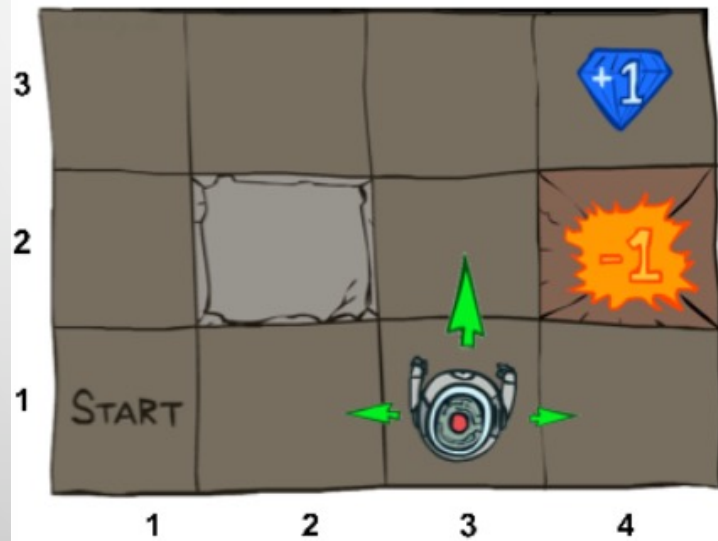
$$\pi_k^*(s) \leftarrow \arg \max_a \sum_{s'} P(s'|s, a) (R(s, a, s') + \gamma V_{k-1}^*(s'))$$

This is called a **value update** or **Bellman update/back-up**

# Value Iteration

$$V_0(s) \leftarrow 0$$

$k = 0$



Noise = 0.2  
Discount = 0.9

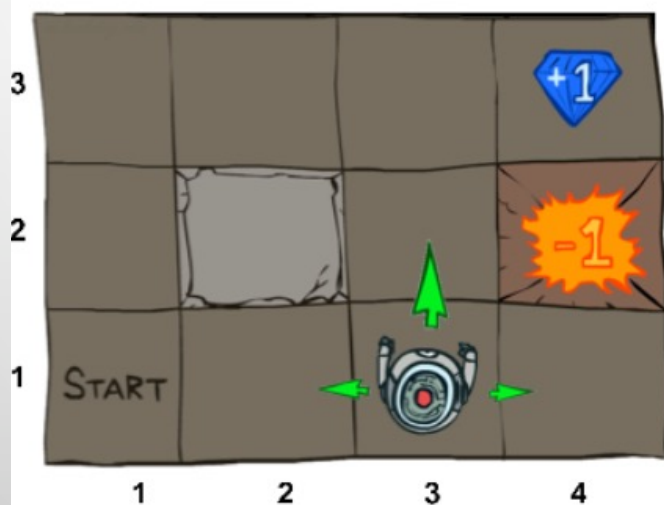
|      |      |      |      |
|------|------|------|------|
| 0.00 | 0.00 | 0.00 | 0.00 |
| 0.00 |      | 0.00 | 0.00 |
| 0.00 | 0.00 | 0.00 | 0.00 |

VALUES AFTER 0 ITERATIONS

# Value Iteration

$$V_1(s) \leftarrow \max_a \sum_{s'} P(s'|s, a) (R(s, a, s') + \gamma V_0(s'))$$

k = 0



Noise = 0.2

Discount = 0.9

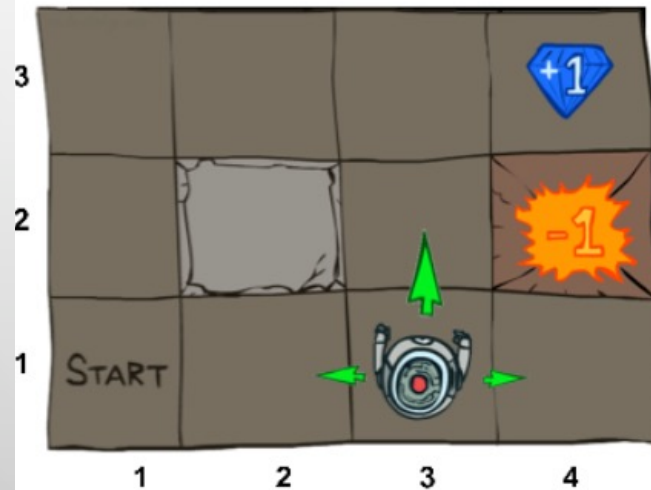
|      |      |      |      |
|------|------|------|------|
| 0.00 | 0.00 | 0.00 | 0.00 |
| 0.00 |      | 0.00 | 0.00 |
| 0.00 | 0.00 | 0.00 | 0.00 |

VALUES AFTER 0 ITERATIONS

# Value Iteration

$$V_2(s) \leftarrow \max_a \sum_{s'} P(s'|s, a) (R(s, a, s') + \gamma V_1(s'))$$

k = 1



|      |      |      |       |
|------|------|------|-------|
| 0.00 | 0.00 | 0.00 | 1.00  |
| 0.00 |      | 0.00 | -1.00 |
| 0.00 | 0.00 | 0.00 | 0.00  |

VALUES AFTER 1 ITERATIONS

Noise = 0.2  
Discount = 0.9

# Value Iteration

$$V_2(s) \leftarrow \max_a \sum_{s'} P(s'|s, a) (R(s, a, s') + \gamma V_1(s'))$$

k = 2

|      |      |      |       |
|------|------|------|-------|
| 0.00 | 0.00 | 0.72 | 1.00  |
| 0.00 |      | 0.00 | -1.00 |
| 0.00 | 0.00 | 0.00 | 0.00  |

VALUES AFTER 2 ITERATIONS

Noise = 0.2  
Discount = 0.9



# Value Iteration

$$V_{k+1}(s) \leftarrow \max_a \sum_{s'} P(s'|s, a) (R(s, a, s') + \gamma V_k(s'))$$

k = 3

|      |      |      |       |
|------|------|------|-------|
| 0.00 | 0.52 | 0.78 | 1.00  |
| 0.00 |      | 0.43 | -1.00 |
| 0.00 | 0.00 | 0.00 | 0.00  |

VALUES AFTER 3 ITERATIONS

Noise = 0.2

Discount = 0.9

# Value Iteration

$$V_{k+1}(s) \leftarrow \max_a \sum_{s'} P(s'|s, a) (R(s, a, s') + \gamma V_k(s'))$$

k = 4

|      |      |      |       |
|------|------|------|-------|
| 0.37 | 0.66 | 0.83 | 1.00  |
| 0.00 |      | 0.51 | -1.00 |
| 0.00 | 0.00 | 0.31 | 0.00  |

VALUES AFTER 4 ITERATIONS

Noise = 0.2

Discount = 0.9

# Value Iteration

$$V_{k+1}(s) \leftarrow \max_a \sum_{s'} P(s'|s, a) (R(s, a, s') + \gamma V_k(s'))$$

k = 5

|      |      |      |       |
|------|------|------|-------|
| 0.51 | 0.72 | 0.84 | 1.00  |
| 0.27 |      | 0.55 | -1.00 |
| 0.00 | 0.22 | 0.37 | 0.13  |

VALUES AFTER 5 ITERATIONS

Noise = 0.2  
Discount = 0.9

# Value Iteration

$$V_{k+1}(s) \leftarrow \max_a \sum_{s'} P(s'|s, a) (R(s, a, s') + \gamma V_k(s'))$$

k = 6

|      |      |      |       |
|------|------|------|-------|
| 0.59 | 0.73 | 0.85 | 1.00  |
| 0.41 |      | 0.57 | -1.00 |
| 0.21 | 0.31 | 0.43 | 0.19  |

VALUES AFTER 6 ITERATIONS

Noise = 0.2  
Discount = 0.9

# Value Iteration

$$V_{k+1}(s) \leftarrow \max_a \sum_{s'} P(s'|s, a) (R(s, a, s') + \gamma V_k(s'))$$

k = 7

|      |      |      |       |
|------|------|------|-------|
| 0.62 | 0.74 | 0.85 | 1.00  |
| 0.50 |      | 0.57 | -1.00 |
| 0.34 | 0.36 | 0.45 | 0.24  |

VALUES AFTER 7 ITERATIONS

Noise = 0.2  
Discount = 0.9

# Value Iteration

$$V_{k+1}(s) \leftarrow \max_a \sum_{s'} P(s'|s, a) (R(s, a, s') + \gamma V_k(s'))$$

k = 8

|      |      |      |       |
|------|------|------|-------|
| 0.63 | 0.74 | 0.85 | 1.00  |
| 0.53 |      | 0.57 | -1.00 |
| 0.42 | 0.39 | 0.46 | 0.26  |

VALUES AFTER 8 ITERATIONS

Noise = 0.2  
Discount = 0.9

# Value Iteration

$$V_{k+1}(s) \leftarrow \max_a \sum_{s'} P(s'|s, a) (R(s, a, s') + \gamma V_k(s'))$$

k = 9



Noise = 0.2  
Discount = 0.9

# Value Iteration

$$V_{k+1}(s) \leftarrow \max_a \sum_{s'} P(s'|s, a) (R(s, a, s') + \gamma V_k(s'))$$

k = 10

|      |      |      |       |
|------|------|------|-------|
| 0.64 | 0.74 | 0.85 | 1.00  |
| 0.56 |      | 0.57 | -1.00 |
| 0.48 | 0.41 | 0.47 | 0.27  |

VALUES AFTER 10 ITERATIONS

Noise = 0.2  
Discount = 0.9



# Value Iteration

$$V_{k+1}(s) \leftarrow \max_a \sum_{s'} P(s'|s, a) (R(s, a, s') + \gamma V_k(s'))$$

k = 11

|      |      |      |       |
|------|------|------|-------|
| 0.64 | 0.74 | 0.85 | 1.00  |
| 0.56 |      | 0.57 | -1.00 |
| 0.48 | 0.42 | 0.47 | 0.27  |

VALUES AFTER 11 ITERATIONS

Noise = 0.2  
Discount = 0.9

# Value Iteration

$$V_{k+1}(s) \leftarrow \max_a \sum_{s'} P(s'|s, a) (R(s, a, s') + \gamma V_k(s'))$$

k = 12

|      |      |      |       |
|------|------|------|-------|
| 0.64 | 0.74 | 0.85 | 1.00  |
| 0.57 |      | 0.57 | -1.00 |
| 0.49 | 0.42 | 0.47 | 0.28  |

VALUES AFTER 12 ITERATIONS

Noise = 0.2  
Discount = 0.9

# Value Iteration

$$V_{k+1}(s) \leftarrow \max_a \sum_{s'} P(s'|s, a) (R(s, a, s') + \gamma V_k(s'))$$

k = 100

|      |      |      |       |
|------|------|------|-------|
| 0.64 | 0.74 | 0.85 | 1.00  |
| 0.57 |      | 0.57 | -1.00 |
| 0.49 | 0.43 | 0.48 | 0.28  |

VALUES AFTER 100 ITERATIONS

Noise = 0.2

Discount = 0.9

# Value Iteration Convergence

**Theorem.** Value iteration converges. At convergence, we have found the optimal value function  $V^*$  for the discounted infinite horizon problem, which satisfies the Bellman equations

$$\forall S \in S : \quad V^*(s) = \max_A \sum_{s'} T(s, a, s') [R(s, a, s') + \gamma V^*(s')]$$

Bellman Optimality Equation

# Proof Sketch (special case)

- Assume  $r \geq 0$
- $V_H(s)$  is a **bounded** and **increasing** sequence in  $H$ .
  - So it converges
- But  $V_{H+1} = \max_a \sum_{s'} P(s'|s, a) [R(s, a, s') + \gamma V_H(s')]$  is a continuous function of  $V_H(s)$ .
- Taking limits of both sides yields the Bellman optimality equation.
- General case: Use **contraction mapping** idea (could be discussed at the recitation class)

# Q-Values

- $Q^*(s, a)$  = expected utility **starting in  $s$ , taking action  $a$ , and (thereafter) acting optimally**

$$V^*(s) = \max_{a'} Q^*(s, a')$$

- Bellman Equation:

$$Q^*(s, a) = \sum_{s'} P(s'|s, a) (R(s, a, s') + \gamma \max_{a'} Q^*(s', a'))$$

- Q-value Iteration:

$$Q_{k+1}^*(s, a) \leftarrow \sum_{s'} P(s'|s, a) (R(s, a, s') + \gamma \max_{a'} Q_k^*(s', a'))$$

# Q-Value Iteration

$$Q_{k+1}^*(s, a) \leftarrow \sum_{s'} P(s'|s, a) (R(s, a, s') + \gamma \max_{a'} Q_k^*(s', a'))$$

k = 100



Noise = 0.2  
Discount = 0.9

# Policy Evaluation

- Recall value iteration:

$$V_k^*(s) \leftarrow \max_a \sum_{s'} P(s'|s, a) (R(s, a, s') + \gamma V_{k-1}^*(s'))$$

- Policy evaluation for a given  $\pi(s)$  :

$$V_k^\pi(s) \leftarrow \sum_{s'} P(s'|s, \pi(s)) (R(s, \pi(s), s') + \gamma V_{k-1}^\pi(s))$$

At convergence:

$$\forall s \quad V^\pi(s) \leftarrow \sum_{s'} P(s'|s, \pi(s)) (R(s, \pi(s), s') + \gamma V^\pi(s))$$



# Policy Iteration

- One iteration of policy iteration

- Policy evaluation for current policy  $\pi_k$  :

- Iterate until convergence

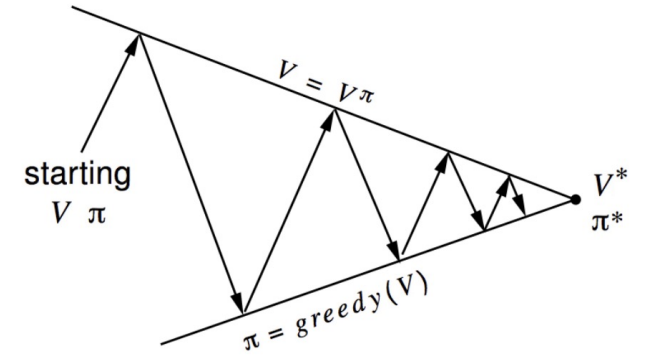
$$V_{i+1}^{\pi_k}(s) \leftarrow \sum_{s'} P(s'|s, \pi_k(s)) [R(s, \pi_k(s), s') + \gamma V_i^{\pi_k}(s')]$$

- Policy improvement: find the best action according to one-step look-ahead

$$\pi_{k+1}(s) \leftarrow \arg \max_a \sum_{s'} P(s'|s, a) [R(s, a, s') + \gamma V^{\pi_k}(s')]$$

- Repeat until policy converges

- At convergence: optimal policy; and converges faster than value iteration under some conditions



# One-step look ahead improves the policy

- Consider an alternative policy  $\pi_{(k+1)}^{(1)}(t, s)$  that takes the prescribed actions in  $\pi_{k+1}(s)$  **only at time  $t = 0$** ; and stays the same as  $\pi_k(s)$  in later times.
- The value function  $V(s)$  for this new policy is larger than or equal to  $V(s)$  for the original policy  $\pi_k(s)$  for all  $s$ . Why?
- Now let  $\pi_{(k+1)}^{(2)}(t, s)$ , which takes the prescribed action in  $\pi_{k+1}(s)$  **only at times  $t = 0$  and  $t = 1$** , and stays the same as  $\pi_k(s)$  in later times.
- Similarly,  $V(s)$  gets improved for  $\pi_{(k+1)}^{(2)}(t, s)$  compared to  $\pi_{(k+1)}^{(1)}(t, s)$  for all  $s$ .
- Repeating this argument  $\pi_{(k+1)}^{(\infty)}(t, s)$  becomes the same as  $\pi_{k+1}(s)$ .

# Policy Iteration Guarantees

Policy Iteration iterates over:

- Policy evaluation

- Iterate until convergence

$$V_{i+1}^{\pi_k}(s) \leftarrow \sum P(s'|s, \pi_k(s)) [R(s, \pi_k(s), s') + \gamma V_i^{\pi_k}(s')]$$

- Policy Improvement

$$\pi_{k+1}(s) \leftarrow \arg \max_a \sum_{s'} P(s'|s, a) [R(s, a, s') + \gamma V^{\pi_k}(s')]$$

**Theorem.** Policy iteration is guaranteed to converge and at convergence, the current policy and its value function are the optimal policy and the optimal value function!

Proof sketch:

- (1) *Guarantee to converge:* In every step the policy improves. This means that a given policy can be encountered at most once. This means that after we have iterated as many times as there are different policies, i.e.,  $(\text{number actions})^{(\text{number states})}$ , we must be done and hence have converged.
- (2) *Optimal at convergence:* by definition of convergence, at convergence  $\pi_{k+1}(s) = \pi_k(s)$  for all states  $s$ . This means  $\forall s \ V^{\pi_k}(s) = \max_a \sum_{s'} T(s, a, s') [R(s, a, s') + \gamma V_i^{\pi_k}(s')]$   
Hence  $V^{\pi_k}$  satisfies the Bellman equation, which means  $V^{\pi_k}$  is equal to the optimal value function  $V^*$ .