# Policy and Value Iteration

# Last Time

# Last Time

- How is a **Markov decision process** defined?

# Last Time

- How is a **Markov decision process** defined?
- What is a **policy**?

# Last Time

- How is a **Markov decision process** defined?
- What is a **policy**?
- How do we **evaluate** policies?

# Last Time

- How is a **Markov decision process** defined?
- What is a **policy**?
- How do we **evaluate** policies?

(MDP notebook)

# Guiding Questions

# Guiding Questions

- How do we reason about the **future consequences** of actions in an MDP?
- What are the basic **algorithms for solving MDPs**?

# Value-Based Policy Evaluation

Discrete, Finite $S$ and $A$

$$U^\pi(s) = E\left[\sum_{t=0}^{\infty} \gamma^t r_t \mid s_0 = s\right]$$

$$= E\left[r_0 \mid s_0 = s\right] + E\left[\sum_{t=1}^{\infty} \gamma^t r_t \mid s_0 = s\right]$$

$$= R(s, \pi(s)) \qquad \qquad ||$$

$$= R(s, \pi(s)) + \sum_{s' \in S} T(s' \mid s, \pi(s))\, E\left[\sum_{t=1}^{\infty} \gamma^t r_t \mid s_1 = s'\right]$$

$\tau = t - 1$

$$= R(s, \pi(s)) + \gamma \sum_{s' \in S} T(s' \mid s, \pi(s))\, \underbrace{E\left[\sum_{\tau=0}^{\infty} \gamma^\tau r_\tau \mid s_0 = s'\right]}_{U^\pi(s')}$$

$$\boxed{U^\pi(s) = R(s, \pi(s)) + \gamma \sum_{s' \in S} T(s' \mid s, \pi(s))\, U^\pi(s')}$$

Bellman's Expectation Eq.

$$U(\pi) = E\left[\sum_{t=0}^{\infty} \gamma^t r_t\right]$$

$$= \underset{s \sim b}{E}\left[U^\pi(s)\right]$$

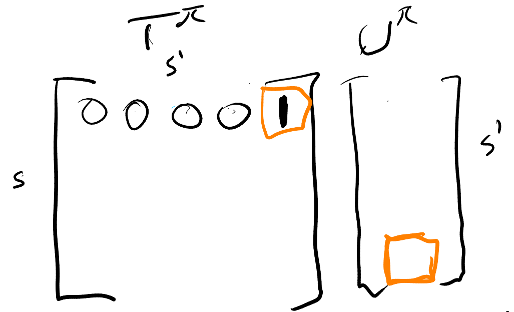
$\vec{U}^\pi \qquad \vec{U}^\pi[\text{ind}(s)] = U^\pi(s)$

$\vec{R}^\pi \qquad \vec{R}^\pi[\text{ind}(s)] = R(s, \pi(s))$

$T^\pi[\text{ind}(s), \text{ind}(s')] = T(s' \mid s, \pi(s))$

$$\vec{U}^\pi = \vec{R}^\pi + \gamma T^\pi \vec{U}^\pi$$

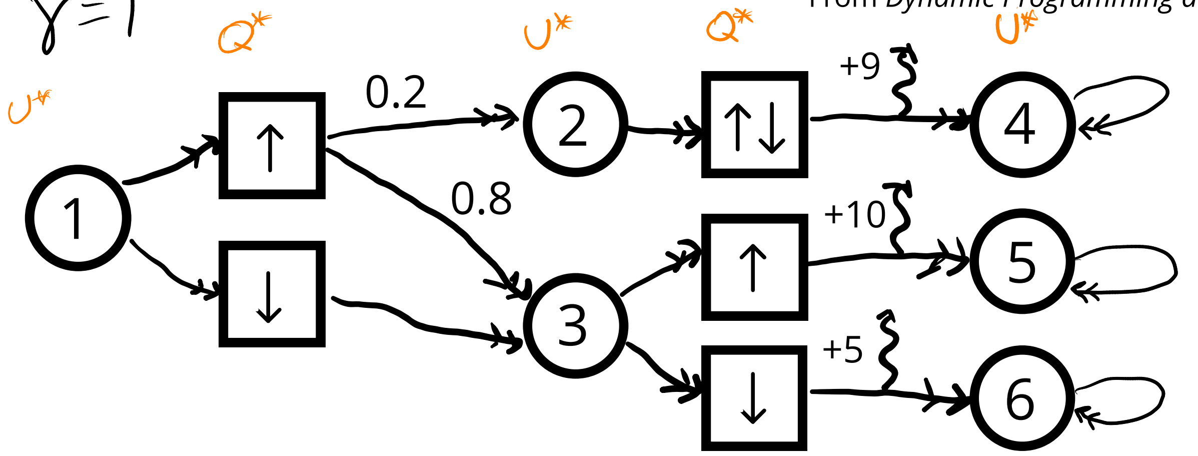
$$\boxed{\vec{U}^\pi = (I - \gamma T^\pi)^{-1} \vec{R}^\pi}$$



# MDP Example: Up-Down Problem

From *Dynamic Programming and the Calculus of Variations*, 1965

$\gamma = 1$



optimal

$U^*(s) = U^{\pi^*}(s)$

$$U^*(s) = \max_a \underbrace{\left(R(s,a) + \gamma \sum_{s' \in S} T(s'|s,a)\, U^*(s')\right)}_{Q^*(s,a)}$$

$U^*(s) = \max_a Q^*(s,a)$

## Algorithm: Bellman Backup

no cycles

Given: MDP $(S, A, R, T, S_T, \gamma)$

→ 1. $U^*(s) \leftarrow 0 \quad \forall s \in S_T$

2. Repeat until $U^*(s)$ known for all states:
    1. Choose $s$ where $U^*$ is known for all children
    2. Calculate $U^*(s)$
3. Extract $\pi^*(s) = \operatorname{argmax} Q^*(s,a)$ ←

| $s$ | $a$ | $Q^*(s,a)$ | $U^*(s)$ |
|---|---|---|---|
| 4 | | | 0 |
| 5 | | | 0 |
| 6 | | | 0 |
| 2 | ↑/↓ | $R(2,\cdot) + 1 \cdot U^*(4)$<br>$+9 + 1 \cdot 0$ | 9 |
| 3 | ↑ | $Q^*(3,\uparrow) = R(3,\uparrow) + 1 \cdot U^*(5)$<br>$= 10 + 0 = \boxed{10}$ | 10 |
| | ↓ | $Q^*(3,\downarrow) = R(3,\downarrow) + 1 \cdot U^*(6)$<br>$5 + 0 = \boxed{5}$ | |
| 1 | ↑ | $Q^*(1,\uparrow) = R(1,\uparrow) + 0.2\, U^*(2) + 0.8\, U^*(3)$<br>$= 0 + 0.2 \cdot 9 + 0.8 \cdot 10$<br>$= \boxed{9.8}$ | 10 |
| | ↓ | $Q^*(1,\downarrow) = 0 + 1 \cdot U^*(3) = \boxed{10}$ | |

# Break: DIA Run

# Policy Iteration

Algorithm: Policy Iteration

Given: MDP $(S, A, R, T, \gamma, b)$

# Policy Iteration

Algorithm: Policy Iteration

Given: MDP $(S, A, R, T, \gamma, b)$

1. initialize $\pi$, $\pi'$ (differently)

# Policy Iteration

Algorithm: Policy Iteration

Given: MDP $(S, A, R, T, \gamma, b)$

```
1. initialize $\pi$, $\pi'$ (differently)
2. while $\pi \neq \pi'$
```

# Policy Iteration

Algorithm: Policy Iteration

Given: MDP $(S, A, R, T, \gamma, b)$

```
1. initialize $\pi$, $\pi'$ (differently)
2. while $\pi \neq \pi'$
3.     $\pi \leftarrow \pi'$
```

# Policy Iteration

Algorithm: Policy Iteration

Given: MDP $(S, A, R, T, \gamma, b)$

```
1. initialize $\pi$, $\pi'$ (differently)
2. while $\pi \neq \pi'$
3.     $\pi \leftarrow \pi'$
4.     $U^\pi \leftarrow (I - \gamma T^\pi)^{-1} R^\pi$
```

# Policy Iteration

Algorithm: Policy Iteration

Given: MDP $(S, A, R, T, \gamma, b)$

```
1. initialize $\pi$, $\pi'$ (differently)
2. while $\pi \neq \pi'$
3.     $\pi \leftarrow \pi'$
4.     $U^\pi \leftarrow (I - \gamma T^\pi)^{-1} R^\pi$
5.     $\pi'(s) \leftarrow \underset{a \in A}{\text{argmax}} \left(R(s, a) + \gamma \sum_{s' \in S} T(s' | s, a) U^\pi(s')\right) \quad \forall s \in S$
```

# Policy Iteration

Algorithm: Policy Iteration

Given: MDP $(S, A, R, T, \gamma, b)$

```
1. initialize $\pi$, $\pi'$ (differently)
2. while $\pi \neq \pi'$
3.     $\pi \leftarrow \pi'$
4.     $U^\pi \leftarrow (I - \gamma T^\pi)^{-1} R^\pi$
5.     $\pi'(s) \leftarrow \underset{a \in A}{\text{argmax}} \left( R(s, a) + \gamma \sum_{s' \in S} T(s' | s, a) U^\pi(s') \right) \quad \forall s \in S$
6. return $\pi$
```

# Policy Iteration

Algorithm: Policy Iteration

Given: MDP $(S, A, R, T, \gamma, b)$

```
initialize $\pi$, $\pi'$ (differently)
while $\pi \neq \pi'$
    $\pi \leftarrow \pi'$
    $U^\pi \leftarrow (I - \gamma T^\pi)^{-1} R^\pi$
    $\pi'(s) \leftarrow \underset{a \in A}{\text{argmax}} \left( R(s,a) + \gamma \sum_{s' \in S} T(s'|s,a) U^\pi(s') \right) \quad \forall s \in S$
return $\pi$
```

(Policy iteration notebook)

# Value Iteration

Algorithm: Value Iteration

Given: MDP $(S, A, R, T, \gamma, b)$, tolerance $\epsilon$

# Value Iteration

Algorithm: Value Iteration

Given: MDP $(S, A, R, T, \gamma, b)$, tolerance $\epsilon$

1. initialize $U$, $U'$ (differently)

# Value Iteration

Algorithm: Value Iteration

Given: MDP $(S, A, R, T, \gamma, b)$, tolerance $\epsilon$

```
1. initialize $U$, $U'$ (differently)
2. while $\|U - U'\|_\infty > \epsilon$
```

# Value Iteration

Algorithm: Value Iteration

Given: MDP $(S, A, R, T, \gamma, b)$, tolerance $\epsilon$

```
initialize $U$, $U'$ (differently)
while $\|U - U'\|_\infty > \epsilon$
    $U \leftarrow U'$
```

# Value Iteration

Algorithm: Value Iteration

Given: MDP $(S, A, R, T, \gamma, b)$, tolerance $\epsilon$

```
initialize $U$, $U'$ (differently)
while $\|U - U'\|_\infty > \epsilon$
    $U \leftarrow U'$
    $U'(s) \leftarrow \max_{a \in A} \left(R(s, a) + \gamma \sum_{s' \in S} T(s' | s, a) U(s')\right) \quad \forall s \in S$
```

# Value Iteration

Algorithm: Value Iteration

Given: MDP $(S, A, R, T, \gamma, b)$, tolerance $\epsilon$

```
1. initialize $U$, $U'$ (differently)
2. while $\|U - U'\|_\infty > \epsilon$
3.     $U \leftarrow U'$
4.     $U'(s) \leftarrow \max_{a \in A} \left(R(s,a) + \gamma \sum_{s' \in S} T(s'|s,a) U(s')\right) \quad \forall s \in S$
5. return $U'$
```

# Value Iteration

Algorithm: Value Iteration

Given: MDP $(S, A, R, T, \gamma, b)$, tolerance $\epsilon$

```
initialize $U$, $U'$ (differently)
while $\|U - U'\|_\infty > \epsilon$
    $U \leftarrow U'$
    $U'(s) \leftarrow \max_{a \in A} \left(R(s,a) + \gamma \sum_{s' \in S} T(s'|s,a) U(s')\right) \quad \forall s \in S$
return $U'$
```

- Returned $U'$ will be close to $U^*$!

# Value Iteration

Algorithm: Value Iteration

Given: MDP $(S, A, R, T, \gamma, b)$, tolerance $\epsilon$

```
initialize $U$, $U'$ (differently)
while $\|U - U'\|_\infty > \epsilon$
    $U \leftarrow U'$
    $U'(s) \leftarrow \max_{a \in A} \left(R(s,a) + \gamma \sum_{s' \in S} T(s'|s,a) U(s')\right) \quad \forall s \in S$
return $U'$
```

- Returned $U'$ will be close to $U^*$!
- $\pi^*$ is easy to extract: $\pi^*(s) = \arg\max(R(s,a) + \gamma E[U^*(s)])$

$$U(\pi) = \mathop{E}_{s\sim b}\left[U^{\pi}(s)\right]$$

$$\mathop{E}_{s'\sim T(s'|s,a)}\left[U(s')\right] = \sum_{s'} T(s'|s,a)\,U(s')$$

# Bellman's Equations

Policy Evaluation

$$U^{\pi}(s) = R(s,\pi(s)) + \gamma \mathop{E}_{s'\sim T(s'|s,a)}\left[U^{\pi}(s')\right]$$

Bellman's Expectation Equation

Certificate of Optimality

Bellman Backup

$$U^{*}(s) = \max_{a}\left(R(s,a) + \gamma \mathop{E}_{s'\sim T(s'|s,a)}\left[U^{*}(s')\right]\right)$$

Bellman's Optimality Equation

Value Iteration

$$U'(s) = \max_{a}\left(R(s,a) + \gamma \mathop{E}_{s'\sim T(s'|s,a)}\left[U(s')\right]\right)$$

Bellman's Operator

$$U'(s) = B[U](s)$$

VI

```
initialize $U$, $U'$
while $\|U - U'\|_\infty > \varepsilon$
    $U \leftarrow U'$
    $U' \leftarrow B[U]$
```

# Guiding Questions

# Guiding Questions

- How do we reason about the **future consequences** of actions in an MDP?
- What are the basic **algorithms for solving MDPs**?

# Guiding Questions

- How do we reason about the **future consequences** of actions in an MDP?
- What are the basic **algorithms for solving MDPs**?

"In any small change he will have to consider only these quantitative indices (or "values") in which all the relevant information is concentrated; and by adjusting the quantities one by one, he can appropriately rearrange his dispositions without having to solve the whole puzzle ab initio, or without needing at any stage to survey it at once in all its ramifications."

-- F. A. Hayek, "The use of knowledge in society", 1945