

---

---

Ayudantía 11

# Reinforcement Learning

Daniel Florea - Daniel Toribio - Pablo Soto

---

---

# Reinforcement Learning

---

# Reinforcement Learning

El problema se modela como un MDP



---

# Markov decision process

MDP se compone:

- Set finito de estados:  $s_1, \dots, s_n$
- Set de recompensas:  $r_1, \dots, r_n$
- Set de acciones:  $a_1, \dots, a_n$
- Set de probabilidades de transición entre estados:

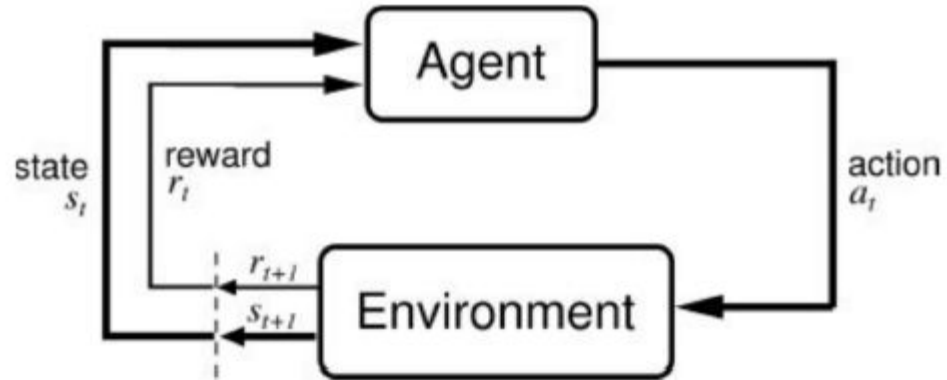
$$P_{ij}^k = P(s_j | s_i, a_k)$$

---

---

# Reinforcement Learning

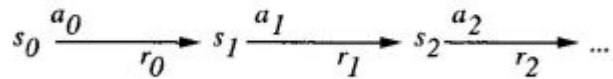
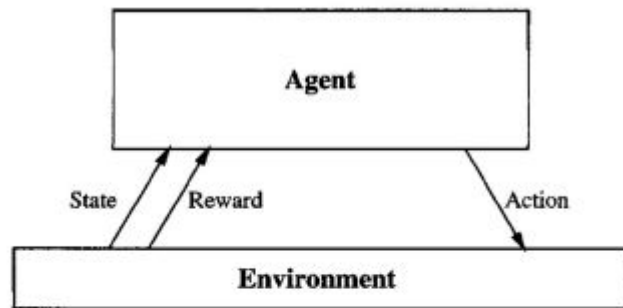
Se busca **aprender a comportarse en el entorno** (aprender una política).



---

# Factor de descuento

Determina la importancia de las **recompensas futuras**



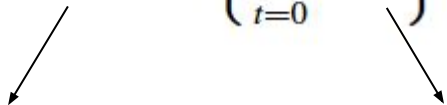
Goal: Learn to choose actions that maximize

$$r_0 + \gamma r_1 + \gamma^2 r_2 + \dots, \text{ where } 0 \leq \gamma < 1$$

---

---

# Value function

$$V(s) = E \left\{ \sum_{t=0}^{\infty} \gamma^t r_t^{\pi} \right\}$$
Two arrows originate from the equation. One arrow points from the state variable 's' in 'V(s)' down to the text 'Recompensa total'. The other arrow points from the circled 'r\_t^{\pi}' term down to the text 'Política de recompensa \pi'.

**Recompensa total** que el agente recibe al iniciar en el estado  $s$  y seguir la política  $\pi$

**Política de recompensa  $\pi$**

---

---

# Value function

- Se busca encontrar la **política  $\pi$  que maximice la *value function*:**

$$V^*(s) = \max_{\pi} E \left\{ \sum_{t=0}^{\infty} \gamma^t r_t^{\pi} \right\}$$

---



---

# Value function

- Se utilizan las ecuaciones de Bellman:

$$V^*(s) = \max_a \left\{ R(s, a) + \gamma \sum_{s' \in S} T(s, a, s') V^*(s') \right\}, \quad \forall s \in S$$

Recompensa del estado actual

Recompensa futura esperada

---

---

# Value function

- Interesan las acciones que maximizan la función

$$V^*(s) = \arg \max_a \left\{ R(s, a) + \gamma \sum_{s' \in S} T(s, a, s') V^*(s') \right\}, \quad \forall s \in S$$

---

—

# Value iteration

---

# Value iteration

```
Initialize  $V(s)$  arbitrarily
loop until policy good enough
  loop for  $s \in S$ 
    loop for  $a \in A$ 
       $Q(s, a) := R(s, a) + \gamma \sum_{s' \in S} T(s, a, s') \hat{V}(s')$ 
    end loop
     $\hat{V}(s) := \max_a Q(s, a)$ 
  end loop
end loop
return  $\{\hat{V}(s)\}$ 
```

---

---

# Value iteration

```
Initialize  $V(s)$  arbitrarily
loop until policy good enough
  loop for  $s \in S$ 
    loop for  $a \in A$ 
       $Q(s, a) := R(s, a) + \gamma \sum_{s' \in S} T(s, a, s') \hat{V}(s')$ 
    end loop
     $\hat{V}(s) := \max_a Q(s, a)$ 
  end loop
end loop
return  $\{\hat{V}(s)\}$ 
```

(2,2)	(3,2)
(2,3)	
(2,4)	

---

# Value iteration

```
Initialize  $V(s)$  arbitrarily
loop until policy good enough
  loop for  $s \in S$ 
    loop for  $a \in A$ 
       $Q(s, a) := R(s, a) + \gamma \sum_{s' \in S} T(s, a, s') \hat{V}(s')$ 
    end loop
     $\hat{V}(s) := \max_a Q(s, a)$ 
  end loop
end loop
return  $\{\hat{V}(s)\}$ 
```

(2,2)	(3,2)
(2,3)	
(2,4)	

---

# Value iteration

$$\gamma = 0.9$$

```
Initialize  $V(s)$  arbitrarily
loop until policy good enough
  loop for  $s \in S$ 
    loop for  $a \in A$ 
       $Q(s, a) := R(s, a) + \gamma \sum_{s' \in S} T(s, a, s') \hat{V}(s')$ 
    end loop
     $\hat{V}(s) := \max_a Q(s, a)$ 
  end loop
end loop
return  $\{\hat{V}(s)\}$ 
```

(2,2)	(3,2)
(2,3)	
(2,4)	

# Value iteration

$$\gamma = 0.9$$

```
Initialize V(s) arbitrarily
loop until policy good enough
  loop for  $s \in S$ 
    loop for  $a \in A$ 
       $Q(s, a) := R(s, a) + \gamma \sum_{s' \in S} T(s, a, s') \hat{V}(s')$ 
    end loop
     $\hat{V}(s) := \max_a Q(s, a)$ 
  end loop
end loop
return  $\{\hat{V}(s)\}$ 
```

(2,2) 0	(3,2) 0
(2,3) 0	
(2,4) 0	



# Value iteration

$$\gamma = 0.9$$

```
Initialize  $V(s)$  arbitrarily
loop until policy good enough
  loop for  $s \in S$ 
    loop for  $a \in A$ 
       $Q(s, a) := R(s, a) + \gamma \sum_{s' \in S} T(s, a, s') \hat{V}(s')$ 
    end loop
     $\hat{V}(s) := \max_a Q(s, a)$ 
  end loop
end loop
return  $\{\hat{V}(s)\}$ 
```

(2,2) 0	(3,2) 0
(2,3) 0	
(2,4) 0	

# Value iteration

$$\gamma = 0.9$$

```
Initialize  $V(s)$  arbitrarily
loop until policy good enough
  loop for  $s \in S$ 
    loop for  $a \in A$ 
       $Q(s, a) := R(s, a) + \gamma \sum_{s' \in S} T(s, a, s') \hat{V}(s')$ 
    end loop
     $\hat{V}(s) := \max_a Q(s, a)$ 
  end loop
end loop
return  $\{\hat{V}(s)\}$ 
```

(2,2) 0	(3,2) 0
(2,3) 0	
(2,4) 0	

# Value iteration

$$\gamma = 0.9$$

```
Initialize  $V(s)$  arbitrarily  
loop until policy good enough  
  loop for  $s \in S$ 
```

```
    loop for  $a \in A$ 
```

$$Q(s, a) := R(s, a) + \gamma \sum_{s' \in S} T(s, a, s') \hat{V}(s')$$

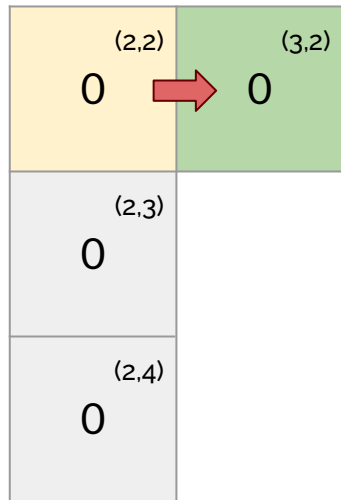
```
    end loop
```

$$\hat{V}(s) := \max_a Q(s, a)$$

```
  end loop
```

```
end loop
```

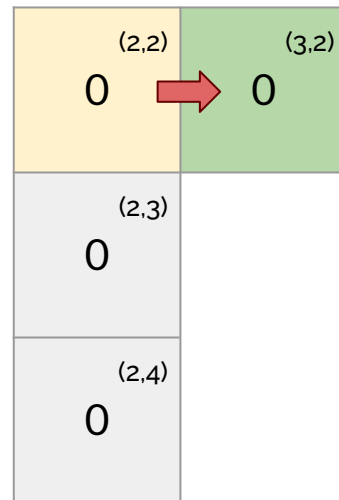
```
return  $\{\hat{V}(s)\}$ 
```



# Value iteration

$$\gamma = 0.9$$

```
Initialize V(s) arbitrarily
loop until policy good enough
  loop for  $s \in S$ 
    loop for  $a \in A$ 
       $Q(s, a) := R(s, a) + \gamma \sum_{s' \in S} T(s, a, s') \hat{V}(s')$ 
    end loop
     $\hat{V}(s) := \max_a Q(s, a)$ 
  end loop
end loop
return  $\{\hat{V}(s)\}$ 
```



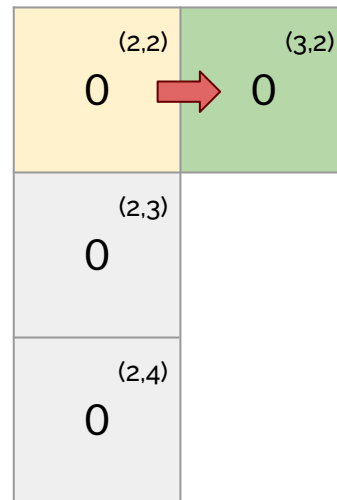
# Value iteration

$$\gamma = 0.9$$

```
Initialize V(s) arbitrarily
loop until policy good enough
  loop for  $s \in S$ 
    loop for  $a \in A$ 
```

$$Q(s, a) := R(s, a) + \gamma \sum_{s' \in S} T(s, a, s') \hat{V}(s')$$

$$Q([2, 2], r) = R([2, 2], r) + \gamma \cdot T([2, 2], r, [3, 2]) \cdot V_0([3, 2])$$



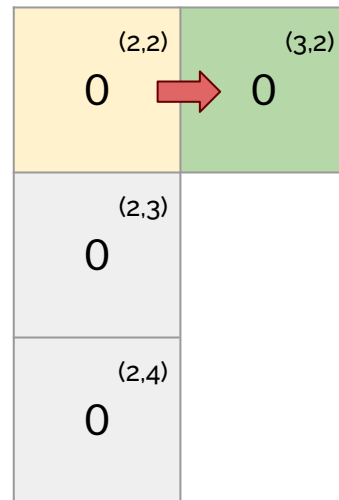
# Value iteration

$$\gamma = 0.9$$

```
Initialize V(s) arbitrarily
loop until policy good enough
  loop for s ∈ S
    loop for a ∈ A
```

$$Q(s, a) := R(s, a) + \gamma \sum_{s' \in S} T(s, a, s') \hat{V}(s')$$

$$\begin{aligned} Q([2, 2], r) &= R([2, 2], r) + \gamma \cdot T([2, 2], r, [3, 2]) \cdot V_0([3, 2]) \\ &= 1 + 0.9 \cdot 0.25 \cdot 0 \end{aligned}$$



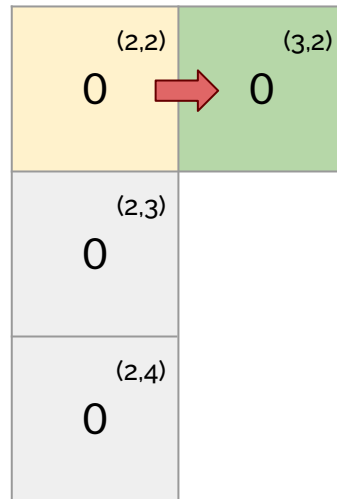
# Value iteration

$$\gamma = 0.9$$

```
Initialize V(s) arbitrarily
loop until policy good enough
  loop for  $s \in S$ 
    loop for  $a \in A$ 
```

$$Q(s, a) := R(s, a) + \gamma \sum_{s' \in S} T(s, a, s') \hat{V}(s')$$

$$\begin{aligned} Q([2, 2], r) &= R([2, 2], r) + \gamma \cdot T([2, 2], r, [3, 2]) \cdot V_0([3, 2]) \\ &= 1 + 0.9 \cdot 0.25 \cdot 0 \\ &= 1, \end{aligned}$$



# Value iteration

$$\gamma = 0.9$$

```
Initialize  $V(s)$  arbitrarily  
loop until policy good enough  
  loop for  $s \in S$   
    loop for  $a \in A$   
       $Q(s, a) := R(s, a) + \gamma \sum_{s' \in S} T(s, a, s') \hat{V}(s')$   
    end loop
```

$$Q([2, 2], r) = 1$$

$$Q([2, 2], l) = 0$$

$$Q([2, 2], u) = 0$$

$$Q([2, 2], d) = 0$$

$(2,2)$ 0	$(3,2)$ 0
$(2,3)$ 0	
$(2,4)$ 0	



# Value iteration

$$\gamma = 0.9$$

```
Initialize  $V(s)$  arbitrarily
loop until policy good enough
  loop for  $s \in S$ 
    loop for  $a \in A$ 
       $Q(s, a) := R(s, a) + \gamma \sum_{s' \in S} T(s, a, s') \hat{V}(s')$ 
    end loop
     $\hat{V}(s) := \max_a Q(s, a)$ 
```

$$Q([2, 2], r) = 1$$

$$Q([2, 2], l) = 0$$

$$Q([2, 2], u) = 0$$

$$Q([2, 2], d) = 0$$

(2,2) 1	(3,2) 0
(2,3) 0	
(2,4) 0	

t = 0

0	0
0	
0	

# Value iteration

$$\gamma = 0.9$$

Initialize  $V(s)$  arbitrarily  
loop until policy good enough

  loop for  $s \in S$

    loop for  $a \in A$

$$Q(s, a) := R(s, a) + \gamma \sum_{s' \in S} T(s, a, s') \hat{V}(s')$$

  end loop

(2,2) 1	(3,2) 0
(2,3) 0	
(2,4) 0	

t = 0

0	0
0	
0	

# Value iteration

$$\gamma = 0.9$$

```
Initialize  $V(s)$  arbitrarily  
loop until policy good enough  
  loop for  $s \in S$   
    loop for  $a \in A$ 
```

$$Q(s, a) := R(s, a) + \gamma \sum_{s' \in S} T(s, a, s') \hat{V}(s')$$

```
  end loop
```

$$Q([3, 2], r) = 1$$

$$Q([3, 2], l) = 0$$

$$Q([3, 2], u) = 1$$

$$Q([3, 2], d) = 1$$

(2,2) 1	(3,2) 0
(2,3) 0	
(2,4) 0	

t = 0

0	0
0	
0	

# Value iteration

$$\gamma = 0.9$$

```
Initialize V(s) arbitrarily
loop until policy good enough
  loop for  $s \in S$ 
    loop for  $a \in A$ 
       $Q(s, a) := R(s, a) + \gamma \sum_{s' \in S} T(s, a, s') \hat{V}(s')$ 
    end loop
  end loop
   $\hat{V}(s) := \max_a Q(s, a)$ 
```

$$Q([3, 2], r) = 1$$

$$Q([3, 2], l) = 0$$

$$Q([3, 2], u) = 1$$

$$Q([3, 2], d) = 1$$

(2,2) 1	(3,2) 1
(2,3) 0	
(2,4) 0	

t = 0

0	0
0	
0	

# Value iteration

$$\gamma = 0.9$$

Initialize  $V(s)$  arbitrarily  
loop until policy good enough

loop for  $s \in S$

loop for  $a \in A$

$$Q(s, a) := R(s, a) + \gamma \sum_{s' \in S} T(s, a, s') \hat{V}(s')$$

end loop

(2,2) 1	(3,2) 1
(2,3) 0	
(2,4) 0	

t = 0

0	0
0	
0	

# Value iteration

$$\gamma = 0.9$$

```
Initialize V(s) arbitrarily
loop until policy good enough
  loop for  $s \in S$ 
    loop for  $a \in A$ 
```

$$Q(s, a) := R(s, a) + \gamma \sum_{s' \in S} T(s, a, s') \hat{V}(s')$$

```
    end loop
```

$$Q([2, 3], r) = 0$$

$$Q([2, 3], l) = 0$$

$$Q([2, 3], u) = 0$$

$$Q([2, 3], d) = 0$$

(2,2) 1	(3,2) 1
(2,3) 0	
(2,4) 0	

t = 0

0	0
0	
0	

# Value iteration

$$\gamma = 0.9$$

```
Initialize  $V(s)$  arbitrarily  
loop until policy good enough  
  loop for  $s \in S$   
    loop for  $a \in A$   
       $Q(s, a) := R(s, a) + \gamma \sum_{s' \in S} T(s, a, s') \hat{V}(s')$   
    end loop  
   $\hat{V}(s) := \max_a Q(s, a)$ 
```

$$Q([2, 3], r) = 0$$

$$Q([2, 3], l) = 0$$

$$Q([2, 3], u) = 0$$

$$Q([2, 3], d) = 0$$

(2,2) 1	(3,2) 1
(2,3) 0	
(2,4) 0	

t = 0

0	0
0	
0	

# Value iteration

$$\gamma = 0.9$$

Initialize  $V(s)$  arbitrarily  
loop until policy good enough

loop for  $s \in S$

loop for  $a \in A$

$$Q(s, a) := R(s, a) + \gamma \sum_{s' \in S} T(s, a, s') \hat{V}(s')$$

end loop

(2,2) 1	(3,2) 1
(2,3) 0	
(2,4) 0	

t = 0

0	0
0	
0	



# Value iteration

$$\gamma = 0.9$$

```
Initialize V(s) arbitrarily  
loop until policy good enough  
  loop for  $s \in S$   
    loop for  $a \in A$ 
```

$$Q(s, a) := R(s, a) + \gamma \sum_{s' \in S} T(s, a, s') \hat{V}(s')$$

```
  end loop
```

$$Q([2, 4], r) = 0$$

$$Q([2, 4], l) = 0$$

$$Q([2, 4], u) = 0$$

$$Q([2, 4], d) = 0$$

(2,2) 1	(3,2) 1
(2,3) 0	
(2,4) 0	

t = 0

0	0
0	
0	

# Value iteration

$$\gamma = 0.9$$

```
Initialize  $V(s)$  arbitrarily  
loop until policy good enough  
  loop for  $s \in S$   
    loop for  $a \in A$   
       $Q(s, a) := R(s, a) + \gamma \sum_{s' \in S} T(s, a, s') \hat{V}(s')$   
    end loop  
   $\hat{V}(s) := \max_a Q(s, a)$ 
```

$$Q([2, 4], r) = 0$$

$$Q([2, 4], l) = 0$$

$$Q([2, 4], u) = 0$$

$$Q([2, 4], d) = 0$$

(2,2) 1	(3,2) 1
(2,3) 0	
(2,4) 0	

t = 0

0	0
0	
0	

# Value iteration

$$\gamma = 0.9$$

```
Initialize  $V(s)$  arbitrarily
loop until policy good enough
  loop for  $s \in S$ 
    loop for  $a \in A$ 
       $Q(s, a) := R(s, a) + \gamma \sum_{s' \in S} T(s, a, s') \hat{V}(s')$ 
    end loop
     $\hat{V}(s) := \max_a Q(s, a)$ 
  end loop
end loop
return  $\{\hat{V}(s)\}$ 
```

(2,2) 1	(3,2) 1
(2,3) 0	
(2,4) 0	

t = 0

0	0
0	
0	

# Value iteration

$$\gamma = 0.9$$

```
Initialize  $V(s)$  arbitrarily  
loop until policy good enough
```

```
  loop for  $s \in S$   
    loop for  $a \in A$   
       $Q(s, a) := R(s, a) + \gamma \sum_{s' \in S} T(s, a, s') \hat{V}(s')$   
    end loop  
     $\hat{V}(s) := \max_a Q(s, a)$   
  end loop  
end loop  
return  $\{\hat{V}(s)\}$ 
```

t = 1

(2,2) 1	(3,2) 1
(2,3) 0	
(2,4) 0	

---

# Value iteration

(...)

---

---

# Value iteration

t -> inf

(2,2) b	(3,2) a
(2,3) c	
(2,4) d	

¿ Qué relación existirá entre a, b, c y d?

---

---

# Value iteration

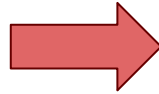
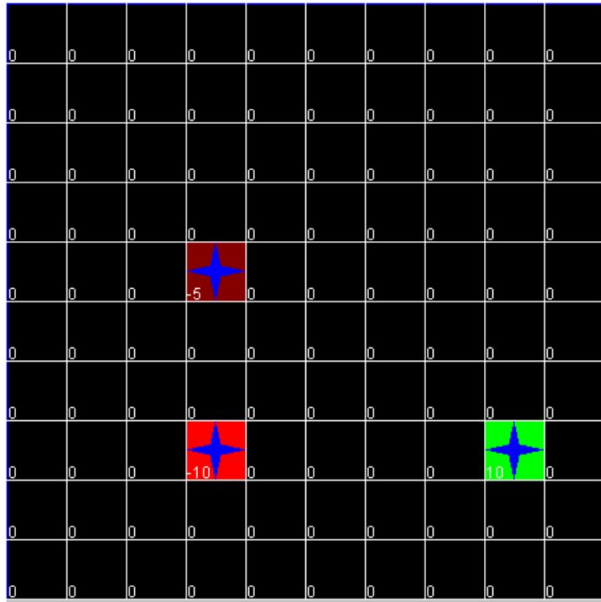
t -> inf

(2,2) b	(3,2) a
(2,3) c	
(2,4) d	

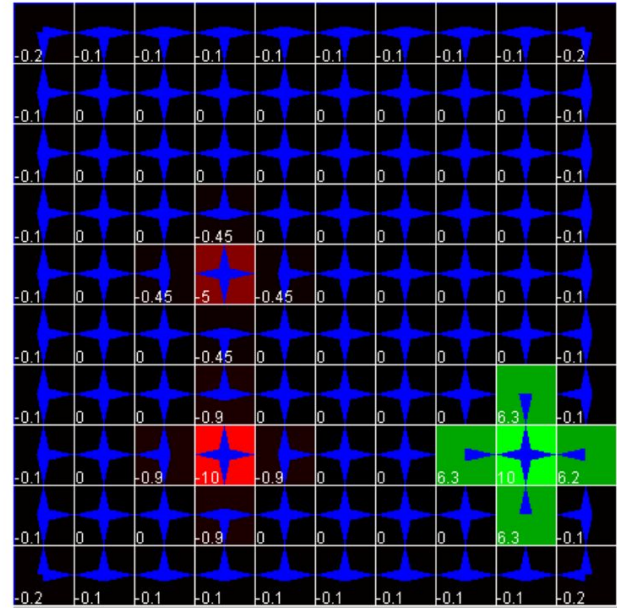
Para este caso en particular

$$a > b > c > d$$

# Value iteration

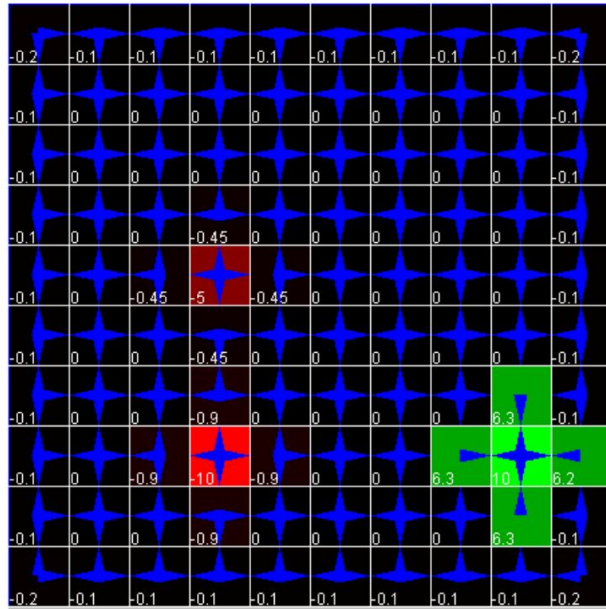


Iteration 1

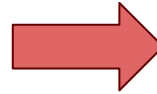




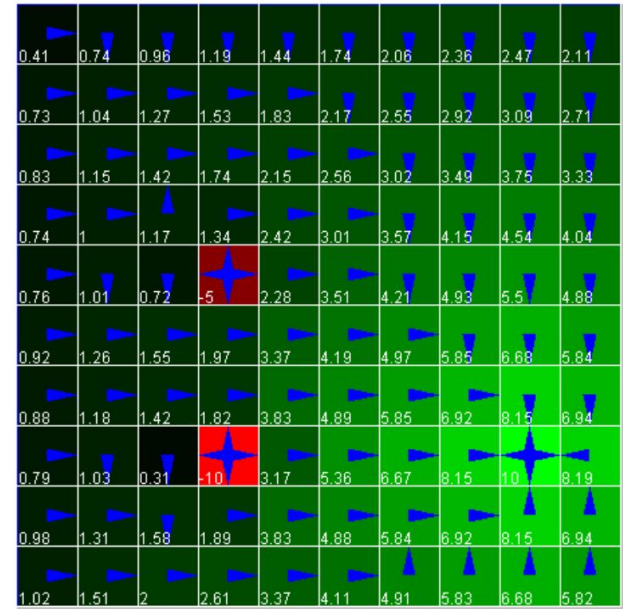
# Value iteration



(...)



## Iteration 10



—

# Policy iteration

---

# Policy iteration

Choose an arbitrary policy  $\pi'$

Loop

$\pi := \pi'$

Compute value function of policy  $\pi$ :

  #solve linear equations

$$V_{\pi}(s) := R(s, \pi(s)) + \gamma \sum_{s' \in S} T(s, \pi(s), s') V_{\pi}(s')$$

Improve the policy at each state

$$\pi'(s) := \arg \max_a (R(s, a) + \gamma \sum_{s' \in S} T(s, a, s') V_{\pi}(s'))$$

until  $\pi = \pi'$

---

---

# Policy iteration

Choose an arbitrary policy  $\pi'$

Loop

$\pi := \pi'$

Compute value function of policy  $\pi$ :

  #solve linear equations

$$V_{\pi}(s) := R(s, \pi(s)) + \gamma \sum_{s' \in S} T(s, \pi(s), s') V_{\pi}(s')$$

Improve the policy at each state

$$\pi'(s) := \arg \max_a (R(s, a) + \gamma \sum_{s' \in S} T(s, a, s') V_{\pi}(s'))$$

until  $\pi = \pi'$

(2,2)	(3,2)
(2,3)	
(2,4)	

# Policy iteration

Choose an arbitrary policy  $\pi'$

Loop

$\pi := \pi'$

Compute value function of policy  $\pi$ :

#solve linear equations

$$V_{\pi}(s) := R(s, \pi(s)) + \gamma \sum_{s' \in S} T(s, \pi(s), s') V_{\pi}(s')$$

Improve the policy at each state

$$\pi'(s) := \arg \max_a (R(s, a) + \gamma \sum_{s' \in S} T(s, a, s') V_{\pi}(s'))$$

until  $\pi = \pi'$

(2,2)	(3,2)
(2,3)	
(2,4)	

Policy (0.3 noise):

- $\pi(2,2) = \text{Right}$
- $\pi(2,3) = \text{Right}$
- $\pi(2,4) = \text{Right}$
- $\pi(3,2) = \text{Stay}$

# Policy iteration

Choose an arbitrary policy  $\pi'$

Loop

$\pi := \pi'$

Compute value function of policy  $\pi$ :

#solve linear equations

$$V_{\pi}(s) := R(s, \pi(s)) + \gamma \sum_{s' \in S} T(s, \pi(s), s') V_{\pi}(s')$$

Improve the policy at each state

$$\pi'(s) := \arg \max_a (R(s, a) + \gamma \sum_{s' \in S} T(s, a, s') V_{\pi}(s'))$$

until  $\pi = \pi'$

(2,2) 0	(3,2) 1
(2,3) 0	
(2,4) 0	

Policy (0.3 noise):

- $\pi(2,2) = \text{Right}$
- $\pi(2,3) = \text{Right}$
- $\pi(2,4) = \text{Right}$
- $\pi(3,2) = \text{Stay}$

# Policy iteration

Choose an arbitrary policy  $\pi'$

Loop

$\pi := \pi'$

Compute value function of policy  $\pi$ :

#solve linear equations

$$V_{\pi}(s) := R(s, \pi(s)) + \gamma \sum_{s' \in S} T(s, \pi(s), s') V_{\pi}(s')$$

Improve the policy at each state

$$\pi'(s) := \arg \max_a (R(s, a) + \gamma \sum_{s' \in S} T(s, a, s') V_{\pi}(s'))$$

until  $\pi = \pi'$

(2,2) 0	(3,2) 1
(2,3) 0	
(2,4) 0	

Policy (0.3 noise):

- $\pi(2,2) = \text{Right}$
- $\pi(2,3) = \text{Right}$
- $\pi(2,4) = \text{Right}$
- $\pi(3,2) = \text{Stay}$

# Policy iteration

Choose an arbitrary policy  $\pi'$

Loop

$\pi := \pi'$

Compute value function of policy  $\pi$ :

#solve linear equations

$$V_{\pi}(s) := R(s, \pi(s)) + \gamma \sum_{s' \in S} T(s, \pi(s), s') V_{\pi}(s')$$

Improve the policy at each state

$$\pi'(s) := \arg \max_a (R(s, a) + \gamma \sum_{s' \in S} T(s, a, s') V_{\pi}(s'))$$

until  $\pi = \pi'$

(2,2) 0	(3,2) 1
(2,3) 0	
(2,4) 0	

Policy (0.3 noise):

- $\pi(2,2) = \text{Right}$
- $\pi(2,3) = \text{Right}$
- $\pi(2,4) = \text{Right}$
- $\pi(3,2) = \text{Stay}$



# Policy iteration

Choose an arbitrary policy  $\pi'$

Loop

$\pi := \pi'$

Compute value function of policy  $\pi$ :

#solve linear equations

$$V_{\pi}(s) := R(s, \pi(s)) + \gamma \sum_{s' \in S} T(s, \pi(s), s') V_{\pi}(s')$$

Improve the policy at each state

$$\pi'(s) := \arg \max_a (R(s, a) + \gamma \sum_{s' \in S} T(s, a, s') V_{\pi}(s'))$$

until  $\pi = \pi'$

(2,2) 0	(3,2) 1
(2,3) 0	
(2,4) 0	

Policy (0.3 noise):

- $\pi(2,2) = \text{Right}$
- $\pi(2,3) = \text{Right}$
- $\pi(2,4) = \text{Right}$
- $\pi(3,2) = \text{Stay}$

# Policy iteration

$$\gamma = 0.9$$

$$V_{\pi}(s) := R(s, \pi(s)) + \gamma \sum_{s' \in S} T(s, \pi(s), s') V_{\pi}(s')$$

$$\begin{aligned} V([2, 2]) = & R([2, 2], \pi) + \gamma( \\ & T([2, 2], \pi, [3, 2]) \cdot V([3, 2]) + \\ & T([2, 2], \pi, [2, 2]) \cdot V([2, 2]) + \\ & T([2, 2], \pi, [2, 3]) \cdot V([2, 3]) \\ & ) \end{aligned}$$

(2,2) 0	(3,2) 1
(2,3) 0	
(2,4) 0	

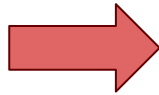
Policy (0.3 noise):

- $\pi(2,2) = \text{Right}$
- $\pi(2,3) = \text{Right}$
- $\pi(2,4) = \text{Right}$
- $\pi(3,2) = \text{Stay}$

# Policy iteration

$$\gamma = 0.9$$

$$V([2, 2]) = R([2, 2], \pi) + \gamma( \\ T([2, 2], \pi, [3, 2]) \cdot V([3, 2]) + \\ T([2, 2], \pi, [2, 2]) \cdot V([2, 2]) + \\ T([2, 2], \pi, [2, 3]) \cdot V([2, 3]) \\ )$$



$$V([2, 2]) = 1 + \gamma( \\ 0,7 \cdot V([3, 2]) + \\ 0,2 \cdot V([2, 2]) + \\ 0,1 \cdot V([2, 3]) \\ )$$

(2,2) 0	(3,2) 1
(2,3) 0	
(2,4) 0	

Policy (0.3 noise):

- $\pi(2,2) = \text{Right}$
- $\pi(2,3) = \text{Right}$
- $\pi(2,4) = \text{Right}$
- $\pi(3,2) = \text{Stay}$

# Policy iteration

$$\gamma = 0.9$$

$$V([2, 2]) = 1 + \gamma(0,7 \cdot V([3, 2]) + 0,2 \cdot V([2, 2]) + 0,1 \cdot V([2, 3]))$$

$$V([3, 2]) = 1 + \gamma(1 \cdot ([3, 2]))$$

$$V([2, 3]) = 0 + \gamma(0,8 \cdot V([2, 3]) + 0,1 \cdot V([2, 2]) + 0,1 \cdot V([2, 4]))$$

$$V([2, 4]) = 0 + \gamma(0,9 \cdot V([2, 4]) + 0,1 \cdot V([2, 3]))$$

(2,2) 0	(3,2) 1
(2,3) 0	
(2,4) 0	

Policy (0.3 noise):

- $\pi(2,2) = \text{Right}$
- $\pi(2,3) = \text{Right}$
- $\pi(2,4) = \text{Right}$
- $\pi(3,2) = \text{Stay}$

# Policy iteration

$$\gamma = 0.9$$

$$V([2, 2]) = 1 + \gamma( \\ 0,7 \cdot V([3, 2]) + \\ 0,2 \cdot V([2, 2]) + \\ 0,1 \cdot V([2, 3]) \\ )$$

$$V([2, 3]) = 0 + \gamma( \\ 0,8 \cdot V([2, 3]) + \\ 0,1 \cdot V([2, 2]) + \\ 0,1 \cdot V([2, 4]) \\ )$$

$$V([2, 4]) = 0 + \gamma( \\ 0,9 \cdot V([2, 4]) + \\ 0,1 \cdot V([2, 3]) \\ )$$

$$V([3, 2]) = 1 + \gamma( \\ 1 \cdot ([3, 2]) \\ )$$

solve

$$a = 1 + 0.9 (0.2 a + 0.1 b + 0.7 d)$$

$$b = 0.9 (0.1 a + 0.8 b + 0.1 c)$$

$$c = 0.9 (0.1 b + 0.9 c)$$

$$d = 1 + 0.9 d$$

(2,2) O	(3,2) 1
(2,3) O	
(2,4) O	

Policy (0.3 noise):

- $\pi(2,2) = \text{Right}$
- $\pi(2,3) = \text{Right}$
- $\pi(2,4) = \text{Right}$
- $\pi(3,2) = \text{Stay}$

# Policy iteration

$$\gamma = 0.9$$

$$V([2, 2]) = 9,289$$

$$V([2, 3]) = 3,522$$

$$V([2, 4]) = 1,668$$

$$V([3, 2]) = 10$$

$(2,2)$ O	$(3,2)$ 1
$(2,3)$ O	
$(2,4)$ O	

Policy (0.3 noise):

- $\pi(2,2) = \text{Right}$
- $\pi(2,3) = \text{Right}$
- $\pi(2,4) = \text{Right}$
- $\pi(3,2) = \text{Stay}$

# Policy iteration

$$\gamma = 0.9$$

$$V([2, 2]) = 9,289$$

$$V([2, 3]) = 3,522$$

$$V([2, 4]) = 1,668$$

$$V([3, 2]) = 10$$

$(2,2)$ 9.289	$(3,2)$ 10
$(2,3)$ 3.522	
$(2,4)$ 1.668	

Policy (0.3 noise):

- $\pi(2,2) = \text{Right}$
- $\pi(2,3) = \text{Right}$
- $\pi(2,4) = \text{Right}$
- $\pi(3,2) = \text{Stay}$

# Policy iteration

$$\gamma = 0.9$$

Choose an arbitrary policy  $\pi'$

Loop

$\pi := \pi'$

Compute value function of policy  $\pi$ :

#solve linear equations

$$V_{\pi}(s) := R(s, \pi(s)) + \gamma \sum_{s' \in S} T(s, \pi(s), s') V_{\pi}(s')$$

Improve the policy at each state

$$\pi'(s) := \arg \max_a (R(s, a) + \gamma \sum_{s' \in S} T(s, a, s') V_{\pi}(s'))$$

until  $\pi = \pi'$

(2,2) 9.289	(3,2) 10
(2,3) 3.522	
(2,4) 1.668	

Policy (0.3 noise):

- $\pi(2,2) = \text{Right}$
- $\pi(2,3) = \text{Right}$
- $\pi(2,4) = \text{Right}$
- $\pi(3,2) = \text{Stay}$



# Policy iteration

$$\gamma = 0.9$$

Choose an arbitrary policy  $\pi'$

Loop

$\pi := \pi'$

Compute value function of policy  $\pi$ :

#solve linear equations

$$V_{\pi}(s) := R(s, \pi(s)) + \gamma \sum_{s' \in S} T(s, \pi(s), s') V_{\pi}(s')$$

Improve the policy at each state

$$\pi'(s) := \arg \max_a (R(s, a) + \gamma \sum_{s' \in S} T(s, a, s') V_{\pi}(s'))$$

until  $\pi = \pi'$

(2,2) 9.289	(3,2) 10
(2,3) 3.522	
(2,4) 1.668	

Policy (0.3 noise):

- $\pi(2,2) = \text{Right}$
- $\pi(2,3) = \text{Right}$
- $\pi(2,4) = \text{Right}$
- $\pi(3,2) = \text{Stay}$

# Policy iteration

$$\gamma = 0.9$$

Choose an arbitrary policy  $\pi'$

Loop

$\pi := \pi'$

Compute value function of policy  $\pi$ :

#solve linear equations

$$V_{\pi}(s) := R(s, \pi(s)) + \gamma \sum_{s' \in \mathcal{S}} T(s, \pi(s), s') V_{\pi}(s')$$

Improve the policy at each state

$$\pi'(s) := \arg \max_a (R(s, a) + \gamma \sum_{s' \in \mathcal{S}} T(s, a, s') V_{\pi}(s'))$$

until  $\pi = \pi'$

(2,2) 9.289	(3,2) 10
(2,3) 3.522	
(2,4) 1.668	

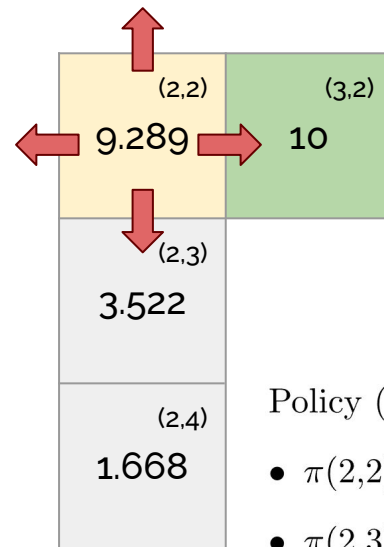
Policy (0.3 noise):

- $\pi(2,2) = \text{Right}$
- $\pi(2,3) = \text{Right}$
- $\pi(2,4) = \text{Right}$
- $\pi(3,2) = \text{Stay}$

# Policy iteration

$$\gamma = 0.9$$

$$R(s, a) + \gamma \sum_{s' \in \mathcal{S}} T(s, a, s') V_{\pi}(s')$$



Policy (0.3 noise):

- $\pi(2,2) = \text{Right}$
- $\pi(2,3) = \text{Right}$
- $\pi(2,4) = \text{Right}$
- $\pi(3,2) = \text{Stay}$

# Policy iteration

$$\gamma = 0.9$$

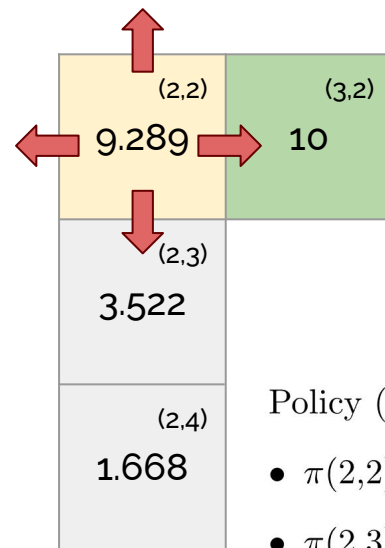
$$R(s, a) + \gamma \sum_{s' \in \mathcal{S}} T(s, a, s') V_{\pi}(s')$$

$$right \rightarrow 1 + 0,9 \cdot (0,7 \cdot 10 + 0,2 \cdot 9,289 + 0,1 \cdot 3,522) = 9,289$$

$$left \rightarrow 0 + 0,9 \cdot (0,8 \cdot 9,289 + 0,1 \cdot 10 + 0,1 \cdot 3,522) = 7,90506$$

$$up \rightarrow 0 + 0,9 \cdot (0,8 \cdot 9,289 + 0,1 \cdot 10 + 0,1 \cdot 3,522) = 7,90506$$

$$down \rightarrow 0 + 0,9 \cdot (0,7 \cdot 3,522 + 0,2 \cdot 9,289 + 0,1 \cdot 10) = 4,79088$$



Policy (0.3 noise):

- $\pi(2,2) = \text{Right}$
- $\pi(2,3) = \text{Right}$
- $\pi(2,4) = \text{Right}$
- $\pi(3,2) = \text{Stay}$

# Policy iteration

$$\gamma = 0.9$$

$$\arg \max_a (R(s, a) + \gamma \sum_{s' \in \mathcal{S}} T(s, a, s') V_{\pi}(s'))$$

$$right \rightarrow 1 + 0,9 \cdot (0,7 \cdot 10 + 0,2 \cdot 9,289 + 0,1 \cdot 3,522) = 9,289$$

$$left \rightarrow 0 + 0,9 \cdot (0,8 \cdot 9,289 + 0,1 \cdot 10 + 0,1 \cdot 3,522) = 7,90506$$

$$up \rightarrow 0 + 0,9 \cdot (0,8 \cdot 9,289 + 0,1 \cdot 10 + 0,1 \cdot 3,522) = 7,90506$$

$$down \rightarrow 0 + 0,9 \cdot (0,7 \cdot 3,522 + 0,2 \cdot 9,289 + 0,1 \cdot 10) = 4,79088$$

(2,2) 9.289	(3,2) 10
(2,3) 3.522	
(2,4) 1.668	

Policy (0.3 noise):

- $\pi(2,2) = \text{Right}$
- $\pi(2,3) = \text{Right}$
- $\pi(2,4) = \text{Right}$
- $\pi(3,2) = \text{Stay}$

# Policy iteration

$$\gamma = 0.9$$

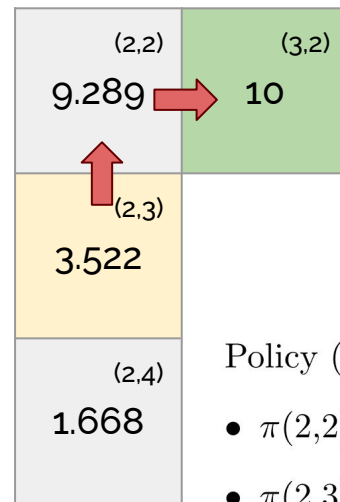
$$\arg \max_a (R(s, a) + \gamma \sum_{s' \in \mathcal{S}} T(s, a, s') V_{\pi}(s'))$$

$$right \rightarrow 0 + 0,9 \cdot (0,8 \cdot 3,522 + 0,1 \cdot 9,289 + 0,1 \cdot 1,668) = 3,52197$$

$$left \rightarrow 0 + 0,9 \cdot (0,8 \cdot 3,522 + 0,1 \cdot 9,289 + 0,1 \cdot 1,668) = 3,52197$$

$$up \rightarrow 1 + 0,9 \cdot (0,7 \cdot 9,289 + 0,2 \cdot 3,522 + 0,1 \cdot 1,668) = 7,63615$$

$$down \rightarrow 0 + 0,9 \cdot (0,7 \cdot 1,668 + 0,2 \cdot 3,522 + 0,1 \cdot 9,289) = 2,52081$$



Policy (0.3 noise):

- $\pi(2,2) = \text{Right}$
- $\pi(2,3) = \text{Right}$
- $\pi(2,4) = \text{Right}$
- $\pi(3,2) = \text{Stay}$

# Policy iteration

$$\gamma = 0.9$$

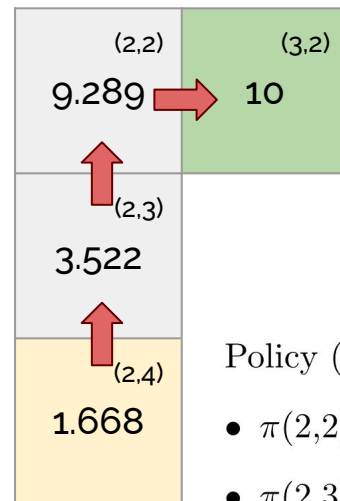
$$\arg \max_a (R(s, a) + \gamma \sum_{s' \in \mathcal{S}} T(s, a, s') V_{\pi}(s'))$$

$$right \rightarrow 0 + 0,9 \cdot (0,9 \cdot 1,668 + 0,1 \cdot 3,522) = 1,66806$$

$$left \rightarrow 0 + 0,9 \cdot (0,9 \cdot 1,668 + 0,1 \cdot 3,522) = 1,66806$$

$$up \rightarrow 1 + 0,9 \cdot (0,7 \cdot 3,522 + 0,3 \cdot 1,668) = 3,66922$$

$$down \rightarrow 0 + 0,9 \cdot (0,9 \cdot 1,668 + 0,1 \cdot 3,522) = 1,66806$$

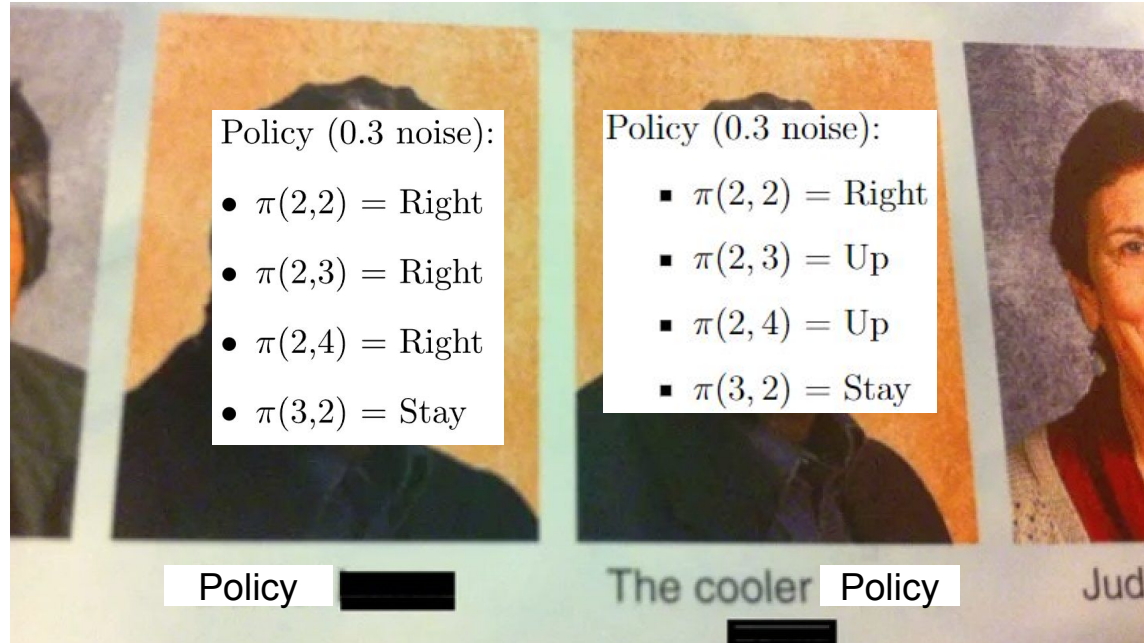


Policy (0.3 noise):

- $\pi(2,2) = \text{Right}$
- $\pi(2,3) = \text{Right}$
- $\pi(2,4) = \text{Right}$
- $\pi(3,2) = \text{Stay}$

---

# Policy iteration





# Policy iteration

$$\gamma = 0.9$$

Choose an arbitrary policy  $\pi'$

Loop

$\pi := \pi'$

Compute value function of policy  $\pi$ :

#solve linear equations

$$V_{\pi}(s) := R(s, \pi(s)) + \gamma \sum_{s' \in S} T(s, \pi(s), s') V_{\pi}(s')$$

Improve the policy at each state

$$\pi'(s) := \arg \max_a (R(s, a) + \gamma \sum_{s' \in S} T(s, a, s') V_{\pi}(s'))$$

until  $\pi = \pi'$

(2,2) 9.289	(3,2) 10
(2,3) 3.522	
(2,4) 1.668	

Policy (0.3 noise):

- $\pi(2, 2) = \text{Right}$
- $\pi(2, 3) = \text{Up}$
- $\pi(2, 4) = \text{Up}$
- $\pi(3, 2) = \text{Stay}$

# Policy iteration

$$\gamma = 0.9$$

Choose an arbitrary policy  $\pi'$

Loop

$\pi := \pi'$

Compute value function of policy  $\pi$ :

#solve linear equations

$$V_{\pi}(s) := R(s, \pi(s)) + \gamma \sum_{s' \in S} T(s, \pi(s), s') V_{\pi}(s')$$

Improve the policy at each state

$$\pi'(s) := \arg \max_a (R(s, a) + \gamma \sum_{s' \in S} T(s, a, s') V_{\pi}(s'))$$

until  $\pi = \pi'$

(2,2) 9.289	(3,2) 10
(2,3) 3.522	
(2,4) 1.668	

Policy (0.3 noise):

- $\pi(2, 2) = \text{Right}$
- $\pi(2, 3) = \text{Up}$
- $\pi(2, 4) = \text{Up}$
- $\pi(3, 2) = \text{Stay}$

---

# Policy iteration

(...)

---

—

# Q-Learning

---

# Q-Learning

---

Q learning algorithm

For each  $s, a$  initialize the table entry  $\hat{Q}(s, a)$  to zero.

Observe the current state  $s$

Do forever:

- Select an action  $a$  and execute it
- Receive immediate reward  $r$
- Observe the new state  $s'$
- Update the table entry for  $\hat{Q}(s, a)$  as follows:

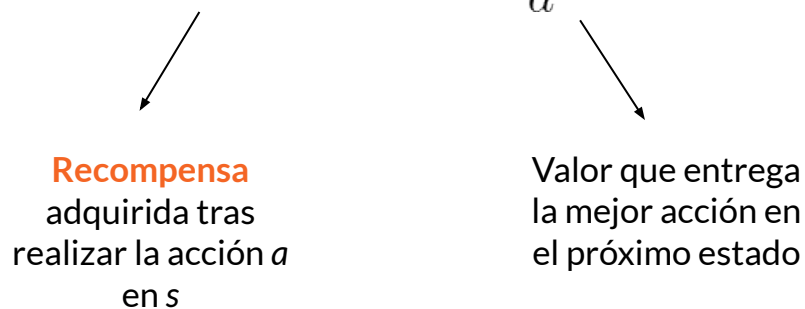
$$\hat{Q}(s, a) \leftarrow r + \gamma \max_{a'} \hat{Q}(s', a')$$

- $s \leftarrow s'$
-

---

# Q-Learning

$$Q(s, a) = r(s, a) + \gamma \arg \max_{a'} Q(s', a')$$



**Recompensa**  
adquirida tras  
realizar la acción  $a$   
en  $s$

Valor que entrega  
la mejor acción en  
el próximo estado

---



Estado	Recompensa
Un grillo	+1
Vacío	-1
Cinco grillos	+10
Pájaro	-10



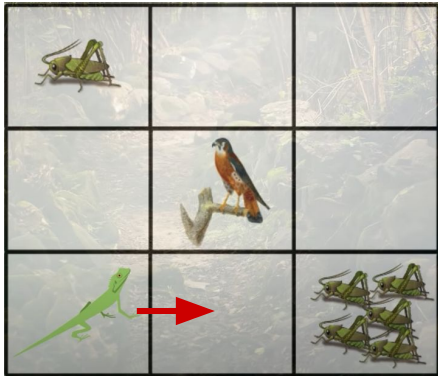
Estado	Recompensa
Un grillo	+1
Vacío	-1
Cinco grillos	+10
Pájaro	-10

---

	Arriba	Abajo	Izq.	Der.
1 grillo	0	0	0	0
Vacío 1	0	0	0	0
Vacío 2	0	0	0	0
Vacío 3	0	0	0	0
Pájaro	0	0	0	0
Vacío 4	0	0	0	0
Vacío 5	0	0	0	0
Vacío 6	0	0	0	0
5 grillos	0	0	0	0

---





Estado	Recompensa
Un grillo	+1
Vacío	-1
Cinco grillos	+10
Pájaro	-10

$$Q(s, a) = \cancel{r(s, a)}^{-1} + \gamma \arg \max_{\cancel{a'}}^0 Q(s', a')$$



Estado	Recompensa
Un grillo	+1
Vacío	-1
Cinco grillos	+10
Pájaro	-10

	Arriba	Abajo	Izq.	Der.
1 grillo	0	0	0	0
Vacío 1	0	0	0	0
Vacío 2	0	0	0	0
Vacío 3	0	0	0	0
Pájaro	0	0	0	0
Vacío 4	0	0	0	0
Vacío 5	0	0	0	-1
Vacío 6	0	0	0	0
5 grillos	0	0	0	0



---

# Q-Learning

---

Q learning algorithm

For each  $s, a$  initialize the table entry  $\hat{Q}(s, a)$  to zero.

Observe the current state  $s$

Do forever:

- Select an action  $a$  and execute it
- Receive immediate reward  $r$
- Observe the new state  $s'$
- Update the table entry for  $\hat{Q}(s, a)$  as follows:

$$\hat{Q}(s, a) \leftarrow r + \gamma \max_{a'} \hat{Q}(s', a')$$

- $s \leftarrow s'$
-

---

# $\epsilon$ - greedy

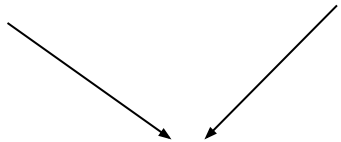
- $\epsilon$ -greedy exploration policy:
    - With probability  $1 - \epsilon$ :
      - Choose the current optimal action:  $\arg \max_a \hat{Q}(s, a)$ .
    - With probability  $\epsilon$ :
      - Select a random action.
-



---

# Q-Learning

$$Q(s, a) = (1-\alpha)Q(s, a) + \alpha(r(s, a) + \gamma \arg \max_{a'} Q(s', a'))$$



**Cuánta**  
**importancia** le doy  
a la experiencia  
nueva

---

---

---

Ayudantía 11

# Reinforcement Learning

**Daniel Florea - Daniel Toribio - Pablo Soto**

---