

# Sistemi Intelligenti Reinforcement Learning: SARSA and Q-learning

Alberto Borghese

Università degli Studi di Milano  
Laboratorio di Sistemi Intelligenti Applicati (AIS-Lab)

Dipartimento di Informatica

[alberto.borghese@unimi.it](mailto:alberto.borghese@unimi.it)

Barto and Sutton, 4.7, 6.4, 6.5



A.A. 2020-2021

1/49

<http://\borghese.di.unimi.it/>



## Sommario



**SARSA**

Q-learning

Esempi

A.A. 2020-2021

2/49

<http://\borghese.di.unimi.it/>



# Meccanismo di apprendimento nel RL



**Inizializzazione:** se l'agente non agisce sull'ambiente non succede nulla. Occorre specificare una policy iniziale.

Ciclo dell'agente (le tre fasi sono sequenziali):

- 1) Implemento una policy ( $\pi(s,a)$ )
- 2) Calcolo la Value function ( $Q^\pi(s,a)$ )
- 3) Aggiorno la policy.

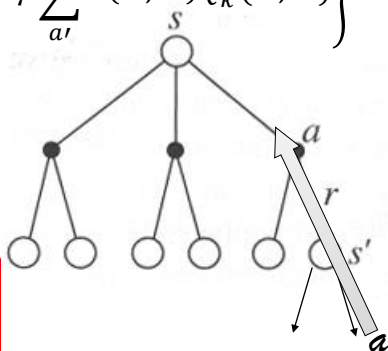
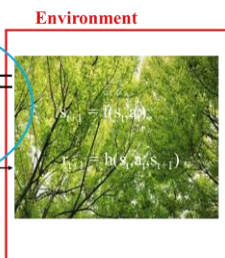
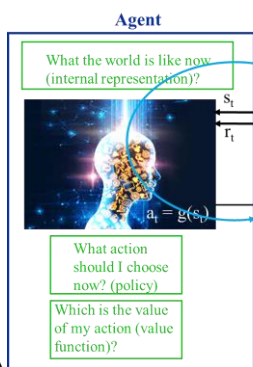


## Un ciclo di interazione



$$Q_{k+1}^\pi(s_t, a_t) = \sum_{s'} P_{s \rightarrow s' | a} \left\{ R_{s,s',a} + \gamma \sum_{a'} \pi(s', a') Q_k^\pi(s', a') \right\}$$

Calcolo ricorsivo di  $Q(\cdot)$



**Passo da t a t+1 poi guardo backwards in time**



# Meccanismo di apprendimento nel RL



**Inizializzazione:** se l'agente non agisce sull'ambiente non succede nulla. Occorre specificare una policy iniziale.

**Ciclo dell'agente (le tre fasi sono sequenziali):**

- 1) Implemento una policy ( $\pi(s,a)$ )
- 2) Determino la Value function ( $Q^\pi(s,a)$ )
- 3) **Aggiorno la policy.**



## $Q(s,a)$ - Osservazioni

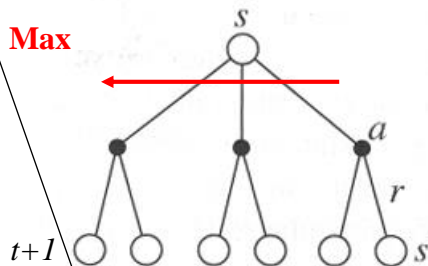


$$Q^\pi(s_t, a_t) = \sum_{s'} P_{s \rightarrow s' | a} \left\{ R_{s,s',a} + \gamma \sum_{a'} \pi(s', a') Q^\pi(s', a') \right\}$$

nota

Per ogni stato devo valutare con informazioni esclusivamente racchiuse in 1 passo l'azione migliore a lungo termine

$$a_{t_{new}} : \max_{a_t} Q^\pi(s_t, a_t)$$



E' supposto noto il funzionamento dell'ambiente (simulazione)



## Effetto del cambiamento della policy



Cambia, a, cambiano i possibili stati successivi ad  $s_t$ ,  $\{s_{t+k}\}$ , ed il reward a lungo termine:

$$Q^\pi(s_t, a_{new}) = E \left\{ r_{t+1} + \gamma \sum_{s'} Pr_{s_t \rightarrow s' | a_{new}} Q_k^\pi(s', a') | s_t = s, a_t = a \right\}$$

$$a_{new} = \pi(s_t) \neq a_t$$

$$Q^{\pi_{new}}(s_t, a_{new}) \stackrel{?}{=} >< Q^\pi(s_t, a_t)$$

Se il reward fosse migliore con  $a_{new}$ , sceglierò sempre  $a_{new}$  in  $s_t$ .

Il reward a lungo termine può essere maggiore (minore) solamente se aumenta (diminuisce) il reward totale "visto" ad un passo (reward del passo + reward successivo).



## Teorema del miglioramento della policy

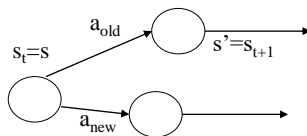




$$Q^\pi(s_t, a_{new}) = E \left\{ r_{t+1} + \gamma \sum_{s'} Pr_{s_t \rightarrow s' | a_{new}} Q_k^\pi(s', a') | s_t = s, a_t = a \right\}$$

**Ipotesi:**  $\pi$  and  $\pi'$  deterministic policies such that  $\forall s_t$

$$Q^{\pi'}(s_t, a_{new} = \pi'(s_t)) \geq Q^\pi(s_t, a_t) \quad \forall a_t$$

**Tesi:**  $\pi'$  è meglio di  $\pi$ . Cioè:  $Q^{\pi'}(s, \pi'(s)) \geq Q^\pi(s, \pi(s)) \quad \forall s$ .



## Learning $Q^\pi(s, a)$

$s_0 = \text{ufficio}; s_5 = \text{casa}.$

$Q_k^\pi(s_0, a_0) = 35$   
 $s_1 \rightarrow Q_{k+1}^\pi(s_0, a_0) = 35$

$Q_k^\pi(s_1, a_1) = 30$   
 $s_2 \rightarrow Q_{k+1}^\pi(s_1, a_1) = 35$

$Q_k^\pi(s_2, a_2) = 15$   
 $s_3 \rightarrow Q_{k+1}^\pi(s_2, a_2) = 20$



$Q_k^\pi(s_3, a_3) = 10$   
 $s_4 \rightarrow Q_{k+1}^\pi(s_3, a_3) = 15$

$Q_k^\pi(s_4, a_4) = 3$   
 $s_5 \rightarrow Q_{k+1}^\pi(s_4, a_4) = 3$

Transitions and rewards:  
 $s_0 \xrightarrow{r_1=5 (5)} s_1$   
 $s_1 \xrightarrow{r_2=20 (15)} s_2$   
 $s_2 \xrightarrow{r_3=10 (5)} s_3$   
 $s_3 \xrightarrow{r_4=12 (7)} s_4$   
 $s_4 \xrightarrow{r_5=3 (3)} s_5$

Come i diversi reward istantanei modificano  $Q^\pi(s, a)$ ?

A.A. 2020-2021 9/49

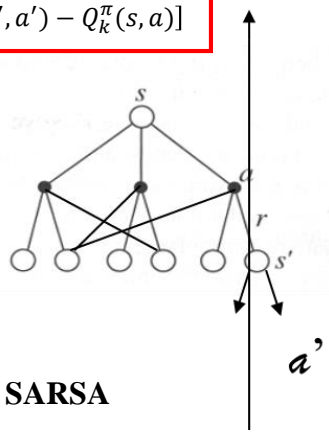



## SARSA

Non richiede conoscenze a priori dell'ambiente.  
 L'agente stima a partire da nulla (bootstrap).  
 Si dimostra che il metodo **converge asintoticamente**, stima  $Q^\pi(s, a)$  quando  $\alpha$  decresce dolcemente a 0 all'aumentare del numero di trial

$$Q_{k+1}^\pi(s, a) = Q_k^\pi(s, a) + \alpha[r' + \gamma Q_k^\pi(s', a') - Q_k^\pi(s, a)]$$

*Sample backup, single state,  
 $s_p$  single action,  $a_p$  single  
 future state  $s' = s_{t+1}$*



**State-Action-Reward-State-Action => SARSA**

A.A. 2020-2021 10/49



# Meccanismo di apprendimento nel RL



**Inizializzazione:** se l'agente non agisce sull'ambiente non succede nulla. Occorre specificare una policy iniziale.

Ciclo dell'agente (le tre fasi sono sequenziali):

- 1) Implemento una policy ( $\pi(s,a)$ )
- 2) **Determino la Value function ( $Q^\pi(s,a)$ )**
- 3) Aggiorno la policy.



# Background su SARSA



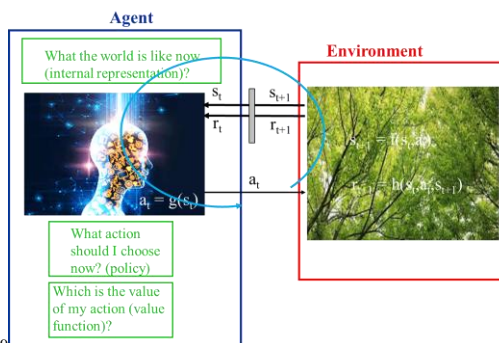
Al tempo  $t$  abbiamo a disposizione:

$r_{t+1} = r'$  estratto (sampled) dalla distribuzione statistica:  $R_{s \rightarrow s' | a_j}$

$s_{t+1} = s'$  estratto (sampled) dalla distribuzione statistica:  $P_{s \rightarrow s' | a_j}$

**Dopo la realizzazione di un evento, l'incertezza statistica scompare.**

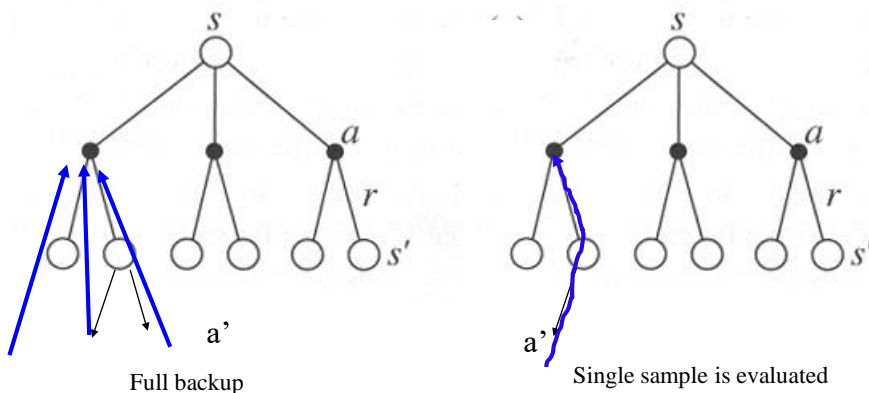
- 1 Reward certo
  - 1 Transizione certa
- vengono forniti dall'ambiente





## Sample backup

$$Q_{k+1}^{\pi}(s, a) = Q_k^{\pi}(s, a) + \alpha[r' + \gamma Q_k^{\pi}(s', a') - Q_k^{\pi}(s, a)]$$



State = s, Action = a, Reward (a un passo) = r, state = s', action = a'

A.A. 2020-2021

13/49



## SARSA Algorithm (progetto)

```

Q(s,a) = rand();           // ∀s, ∀a, eventualmente Q(s,a) = 0 - inizializzazione
Policy definita;        // Policy specificata, eventualmente stocastica
Repeat                    // for each episode
{
  s = s0;
  Repeat                  // for each step of the actual episode
  {
    a = Policy(s);        // policy deterministica o stocastica
    s_next = NextState(s,a); // Funzione non nota all'agente
    reward = Reward(s,s_next,a);
    a_next = Policy(s_next);
    Q(s,a) = Q(s,a) + α [reward + γ Q(s_next, a_next) - Q(s,a)];
    s = s_next;
  } // until last state
} // until the end of learning (convergence of Q(s,a) to true Q(s,a) ∀s, ∀a, for policy π(s,a) )

```

- 1) Apprendiamo il valore di Q **per la policy data (on-policy)**.
- 2) Dopo avere appreso la funzione Q, possiamo modificare la policy in modo da migliorarla. Dovremo poi riapprendere il valore di Q(.)

Come integrare i due passi?

A.A. 2020-2021

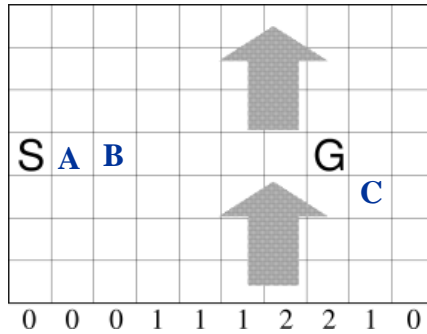
14/49



# Esempio



From Start, S, to Goal, G.



Stati = {caselle della griglia}  
 Stato iniziale = S  
 Terminal state = G  
 Azioni = {su, destra, giù, sinistra}  
 Reward = -1 tranne che quando  $s' = TS$  (reward = 0)

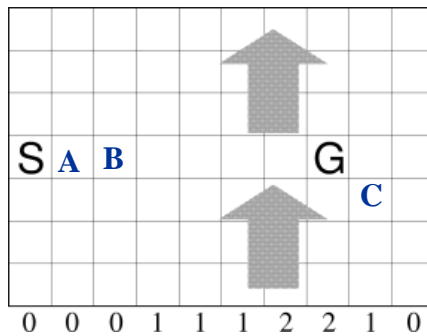
Upwards wind: somma uno spostamento verso l'alto allo spostamento dell'azione dell'agente



# Esempio



From Start, S, to Goal, G.



$Q(s,a)$  iniziale = 0.  
 $r = 0$  se  $s' = G$ ; altrimenti  $r = -1$ .  
 $\pi(s,a)$  data.

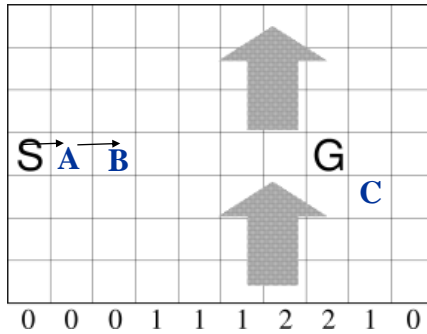
Vogliamo valutare  $\pi(s,a)$ .

$\alpha = 0.5$   
 $\gamma = 1$





## Esempio - risultato



$$\begin{aligned}\epsilon &= 0.1 \\ \alpha &= 0.5 \\ \gamma &= 1\end{aligned}$$

Correzione di Q ad un passo:

$$Q_{k+1}^{\pi}(S, east) = Q_k^{\pi}(S, east) + \alpha[r' + \gamma Q_k^{\pi}(A, east) - Q_k^{\pi}(S, east)] = 0 + 0.5 [-1 + 1 \times 0 - 0] = -0.5$$

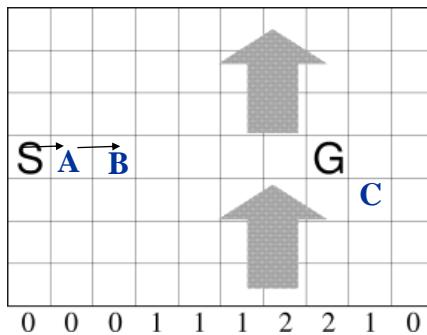
$$Q_{k+1}^{\pi}(s, a) = Q_k^{\pi}(s, a) + \alpha[r' + \gamma Q_k^{\pi}(s', a') - Q_k^{\pi}(s, a)]$$

A.A. 2020-2021

17/49



## Esempio - risultato



$$\begin{aligned}\alpha &= 0.5 \\ \gamma &= 1\end{aligned}$$

Correzione di Q ad un passo:

$$Q_{k+1}^{\pi}(A, east) = Q_k^{\pi}(A, east) + \alpha[r' + \gamma Q_k^{\pi}(B, east) - Q_k^{\pi}(A, east)] = 0 + 0.5 [-1 + 1 \times 0 - 0] = -0.5$$

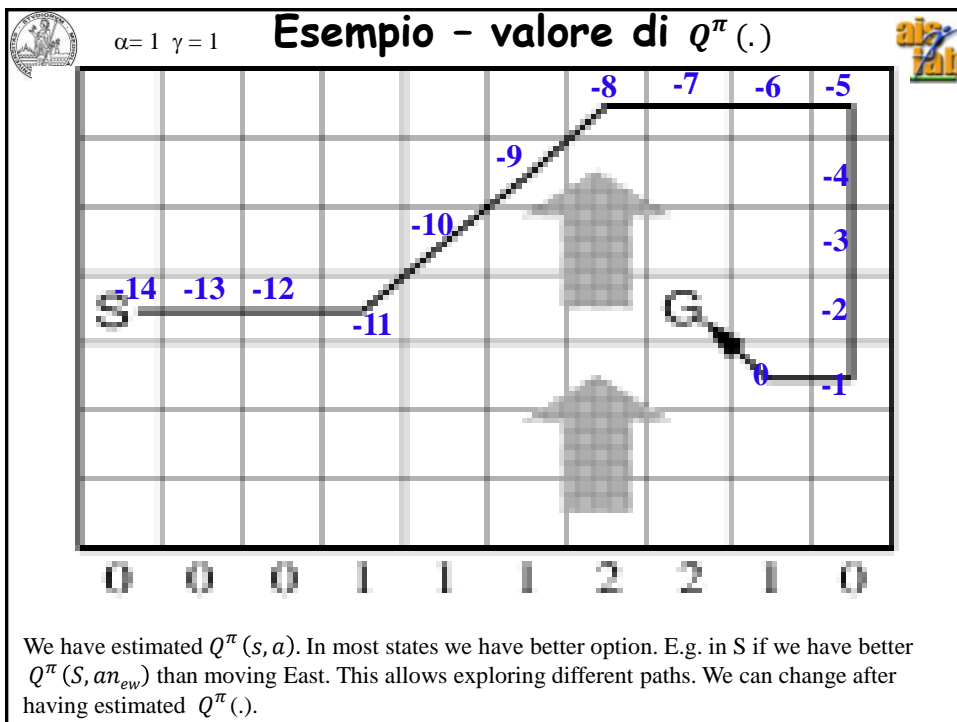
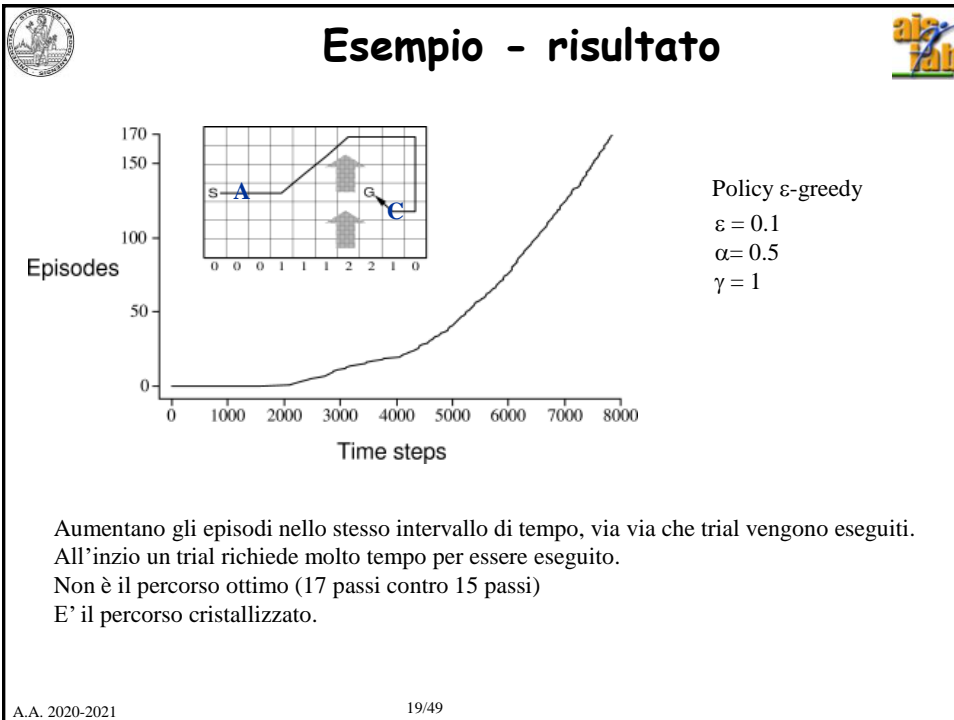
$$Q_{k+1}^{\pi}(C, west) = Q_k^{\pi}(C, west) + \alpha[r' + \gamma Q_k^{\pi}(G, .) - Q_k^{\pi}(C, west)] = 0 + 0.5 [0 + 1 \times 0 - 0] = 0$$

(NB c'è il vento verso l'alto di 1)

$$Q_{k+1}^{\pi}(s, a) = Q_k^{\pi}(s, a) + \alpha[r' + \gamma Q_k^{\pi}(s', a') - Q_k^{\pi}(s, a)]$$

A.A. 2020-2021

18/49





## Sommario



SARSA

Q-learning

Esempi

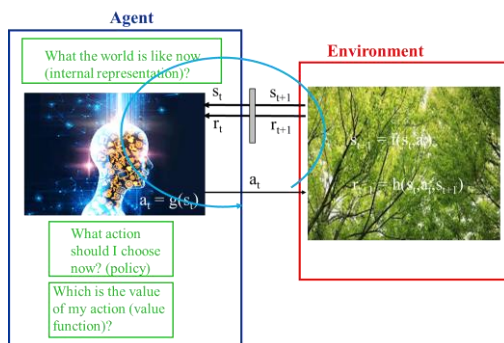


## Value Function?



La Value Function deriva dalla visione della Programmazione Dinamica.

Ma è proprio necessario conoscere esattamente la Value function?  
In fondo a noi interessa determinare la Policy.





# La policy in SARSA

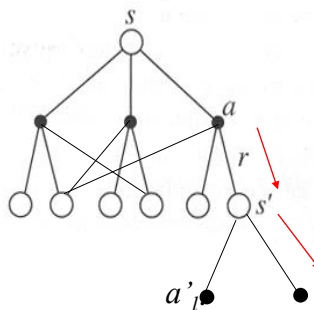


$$Q_{k+1}^\pi(s, a) = Q_k^\pi(s, a) + \alpha[r' + \gamma Q_k^\pi(s', a') - Q_k^\pi(s, a)]$$

1) Apprendiamo il valore di  $Q^\pi(\cdot)$  per una policy data (*on-policy*).

2) Dopo avere appreso la funzione  $Q^\pi(\cdot)$ , possiamo **modificare la policy,  $\pi'(s,a)$** , in modo da migliorarla (**policy improvement**)

3) Dopo avere modificato la policy devo apprendere la nuova  $Q^{\pi'}(\cdot)$



# Un ciclo di interazione



## Agent

What the world is like now  
(internal representation)?

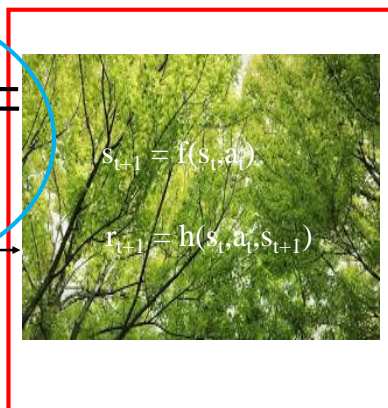


What action  
should I choose  
now? (policy)

Which is the value  
of my action (value  
function)?

$$a_t = g(s_t)$$

## Environment



$$s_{t+1} = f(s_t, a_t)$$

$$r_{t+1} = h(s_t, a_t, s_{t+1})$$

Per ora ci siamo concentrati sulla  
determinazione del valore



# Sommario



SARSA

Q-learning

Esempi



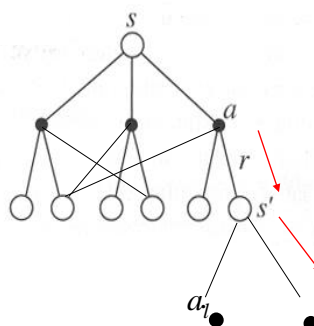
# Come apprendere Q: SARSA



$$Q_{k+1}^{\pi}(s, a) = Q_k^{\pi}(s, a) + \alpha[r' + \gamma Q_k^{\pi}(s', a') - Q_k^{\pi}(s, a)]$$

1) Apprendiamo il valore di Q per una policy data,  $\pi$ , (on-policy).

2) Dopo avere appreso la funzione Q, possiamo modificare la policy in modo da migliorarla (**policy improvement**)



s = state, a = action, r = reward, s = state, a = action

On-policy learning.



## Off-policy Temporal Difference: Q-learning

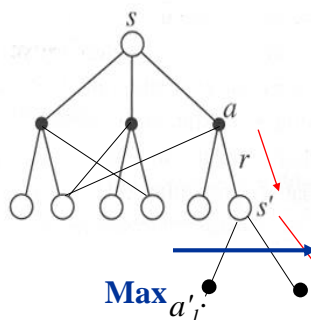


$$Q_{k+1}^{\pi}(s, a) = Q_k^{\pi}(s, a) + \alpha \left[ r' + \gamma \max_{a'} Q_k^{\pi}(s', a') - Q_k^{\pi}(s, a) \right]$$

Non imparo semplicemente la funzione valore  $Q^{\pi}(\cdot)$ , ma la funzione valore  $Q^{\pi^*}(\cdot)$  ottima.

In  $s$ , scelgo un ramo del grafo, e poi **decido** ad un passo come continuare, guardando il reward a lungo termine stimato per le diverse azioni.

Eventualmente cambio subito policy,  $a = \pi(s) \rightarrow a_{\text{new}} = \pi'(s)$  senza aspettare di avere stimato esattamente  $Q^{\pi}(\cdot)$ .



A.A. 2020-2021

27/49

<http://borghese.di.unimi.it/>



## Q-learning algorithm (progetto)



```

Q(s,a) = 0;           // ∀s, ∀a,
Policy data;         // deterministica o stocastica
Repeat
{ s = s0; α = α * reduction_factor; // for each episode
  Repeat // for each step of the single episode
  { a = Policy(s); // eventualmente ε-greedy
    s_next = NextState(s,a); // non nota all'agente
    reward = Reward(s, s_next, a); // non nota all'agente
    a_next_pol = PolicyGreedy(s_next); // on policy (greedy)
    a_next = argmax(Q(s_next, a);
                    a
    if (a_next_pol != a_next) // se esiste un'azione a' migliore
    { UpdatePolicy(s_next, a_next); } // scelgo a_next in s_next da qui in poi
    endif;
    Q(s,a) = Q(s,a) + α [reward + γ Q(s_next, a_next) - Q(s,a)]; // aggiorno Q(s,a)
    s = s_next;
    a = a_next; // a = Policy(s = s_next)
  } // until last state
} // until the end of learning (convergence of Q(s,a) to true Q(s,a) ∀s, ∀a, for policy π(s,a) )

```

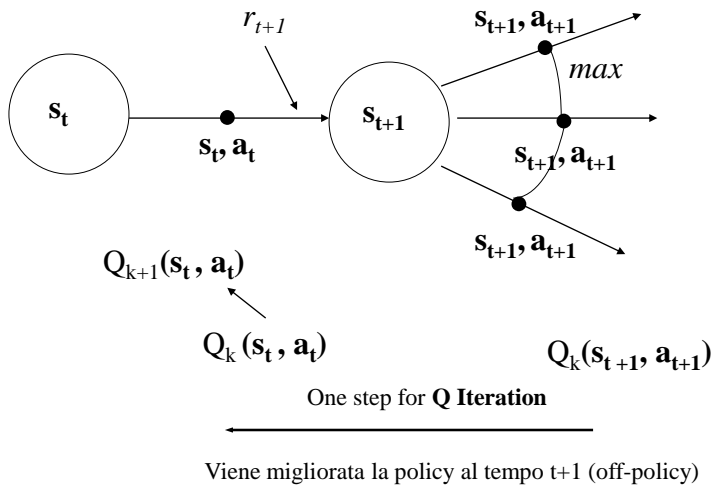
A.A. 2020-2021

28/49

<http://borghese.di.unimi.it/>



## Rappresentazione grafica



A.A. 2020-2021

29/49

<http://borghese.di.unimi.it/>



## Osservazioni

$\pi(s,a)$  sceglie l'azione ottima

$Q^\pi(s, a)$  converge al valore vero (della policy ottima)

Nella pratica la convergenza viene valutata sull'incremento uniforme di Q, ma anche sulla stabilità della policy identificata.

$$Q_{k+1}^\pi(s, a) = Q_k^\pi(s, a) + \alpha \left[ r' + \gamma \max_{a'} Q_k^\pi(s', a') - Q_k^\pi(s, a) \right]$$

L'operazione di max può essere un "hard" max o un "soft" max. Si possono considerare policy  $\epsilon$ -greedy.

A.A. 2020-2021

30/49



## Sommario



SARSA

Q-learning

Esempi



## Example 1 - Q Learning Update



Esempio tratto dai lucidi del corso di Brian C. Williams su RL.

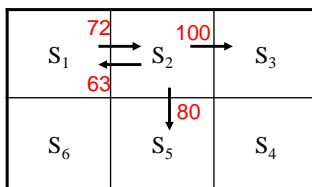
Modificati dalle slide di: Manuela Veloso, Reid Simmons, & Tom Mitchell, CMU

6 stati  $\{s_1, \dots, s_6\}$

Azioni: {su, destra, giù, sinistra}

Reward istantaneo = 0

Inizializzo  $Q(s,a)$  – in rosso.



In rosso i valori di  $Q(s,a)$ .  
Nessun reward istantaneo.

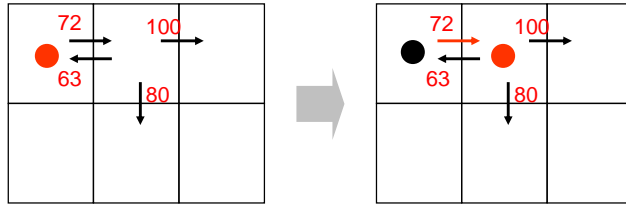




# Example 1 - Q Learning Update



$\gamma = 0.9$   
 $S_{ini} = S_1$



0 reward received in the transition.  $Q(.,.)$  initialized  $\neq 0$

Apprendimento della funzione valore Q. Versione Q-learning.  $Q(s_1, dx) = ?$

$S_1$	$S_2$	$S_3$
$S_6$	$S_5$	$S_4$

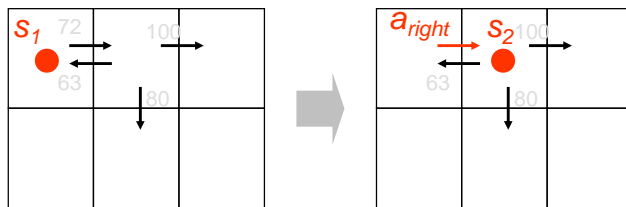
In rosso i valori di  $Q(s,a)$ .  
Nessun reward istantaneo.



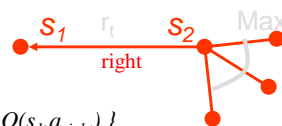
# Example 1 - Q Learning Update



$\gamma = 0.9$   
 $\alpha = 0.1$   
 $a(S_2) = \text{down}$



0 reward received in the transition



$$\begin{aligned}
 \underline{Q}(s_1, a_{right}) &= \underline{Q}(s_1, a_{right}) + \alpha \{ r(s_1, a_{right}, s_2) + \gamma \max_{a'} \underline{Q}(s_2, a') - \underline{Q}(s_1, a_{right}) \} \\
 &= 72 + \alpha \{ 0 + 0.9 \max_{a'} \{ 63, 80, 100 \} - \underline{Q}(s_1, a_{right}) \} \\
 &= 72 + \alpha(90 - 72) = 72 + 1.8 = 73.8
 \end{aligned}$$

Correzione di  $\underline{Q}(s_1, a_{right})$   
 Correzione dell'azione in  $s_2$  da down a right  
 La correzione di  $\underline{Q}(s_1, a_{right})$  va a 0 quando  
 $\underline{Q}(s_1, a_{right}) = 90$

$\underline{Q}(s_2, a_{down}) = 80$   
 $\underline{Q}(s_2, a_{right}) = 100$   
 $\underline{Q}(s_2, a_{left}) = 63$

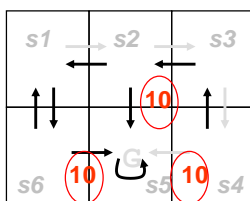


## Example 2: Q-Learning Iterations



- Stati:  $\{s_1, \dots, s_6\}$
- Azioni:  $\{dx, sx, su, giù\}$
- Reward solo in alcune transizioni (in rosso e cerchiato).**
- Stato iniziale:  $s_1$
- Initial selected policy: move clockwise;
- $Q(s,a)$  initially 0;

E.g. videogioco.  
In G rimango in G - loop



$$\alpha = 1$$

$$\gamma = 0.8.$$

A.A. 2020-2021

35/49

<http://borghese.di.unimi.it/>



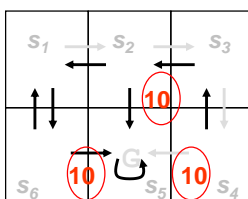
## Example 2: Q-Learning Iterations



- Start at upper left; Initial selected policy: move clockwise;  $Q(s,a)$  initially 0;  $\gamma = 0.8$ .  
**Reward solo nelle transizioni.**

$$Q_{k+1}^{\pi}(s_1, E) = Q_k^{\pi}(s_1, E) + \alpha [r' + \gamma \max_{a'} Q_k^{\pi}(s_2, a') - Q_k^{\pi}(s_1, E)]$$

Reward  
istanteo in  
rosso e  
cerchiato



$$Q_{k+1}^{\pi}(s_1, E) = 0 + 1[0 + 0.8 \times 0 - 0] = 0$$

E.g. videogioco.  
In G rimango in G - loop

$Q(s_1, \text{East})$	$Q(s_2, \text{East})$	$Q(s_3, \text{South})$	$Q(s_4, \text{West})$
0			

A.A. 2020-2021

36/49

<http://borghese.di.unimi.it/>



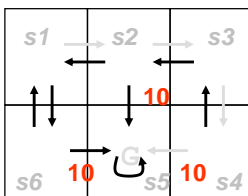
# Q-Learning Iterations - trial 1



- Start at upper left – move clockwise; table initially 0;  $\gamma = 0.8$ ;  $\alpha = 1$

$$Q_{k+1}^\pi(s, a) = Q_k^\pi(s, a) + \alpha \left[ r' + \gamma \max_{a'} Q_k^\pi(s', a') - Q_k^\pi(s, a) \right]$$

$$Q_{k+1}^\pi(s_3, S) = 0 + 1[0 + 0.8 \times 0 - 0] = 0$$



Q(s1,E)	Q(s2,E)	Q(s3,S)	Q(s4,W)
0	0	0	



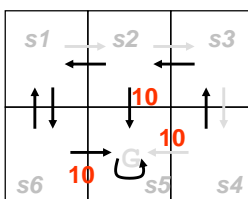
# Q-Learning Iterations - trial 1



- Start at upper left – move clockwise;  $\gamma = 0.8$

$$Q_{k+1}^\pi(s_4, W) = Q_k^\pi(s_4, W) + \alpha \left[ r' + \gamma \max_{a'} Q_k^\pi(s_3, a') - Q_k^\pi(s_4, W) \right]$$

$$Q_{k+1}^\pi(s_4, W) = 0 + 1[10 + 0.8 \times 0 - 0] = 10$$



$Q_k^\pi(s_5, \cdot)$  goal

Q(s1,E)	Q(s2,E)	Q(s3,S)	Q(s4,W)
0	0	0	10



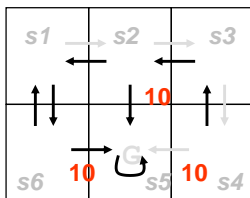
## Q-Learning Iterations - trial 2



- Start at upper left – move clockwise;  $\gamma = 0.8$

$$Q_{k+1}^{\pi}(s_3, S) = Q_k^{\pi}(s_3, S) + \alpha [r' + \gamma \max_{a'} Q_k^{\pi}(s_4, a') - Q_k^{\pi}(s_3, S)]$$

$$Q_{k+1}^{\pi}(s_3, S) = 0 + 1[0 + 0.8 \{ \max, 10, 0 \} - 0] = 8$$



$Q_k^{\pi}(s_4, N)$   
 $Q_k^{\pi}(s_4, W)$

Q(s1,E)	Q(s2,E)	Q(s3,S)	Q(s4,W)
0	0	0	10
0	0	<b>8</b>	

A.A. 2020-2021

39/49

<http://borghese.di.unimi.it/>



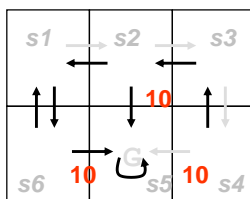
## Q-Learning Iterations - trial 2



- Start at upper left – move clockwise;  $\gamma = 0.8$

$$Q_{k+1}^{\pi}(s_4, W) = Q_k^{\pi}(s_4, W) + \alpha [r' + \gamma \max_{a'} Q_k^{\pi}(s_3, a') - Q_k^{\pi}(s_4, W)]$$

$$Q_{k+1}^{\pi}(s_4, W) = 10 + 1[10 + 0.8 \times 0 - 10] = 10$$



$Q_k^{\pi}(s_3, \cdot)$  goal

Q(s1,E)	Q(s2,E)	Q(s3,S)	Q(s4,W)
0	0	0	10
0	0	8	10

A.A. 2020-2021

40/49

<http://borghese.di.unimi.it/>



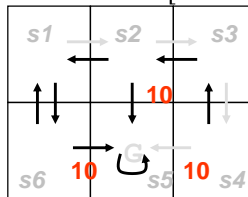
# Q-Learning Iterations - trial 3



- Start at upper left – move clockwise;  $\gamma = 0.8$

$$Q_{k+1}^\pi(s_2, E) = Q_k^\pi(s_2, E) + \alpha [r' + \gamma \max_{a'} Q_k^\pi(s_3, a') - Q_k^\pi(s_2, E)]$$

$$Q_{k+1}^\pi(s_2, E) = 0 + 1 [0 + 0.8 \times \max_{a'} \{8, 0\} - 0] = 6.4$$



$Q_k^\pi(s_3, S)$        $Q_k^\pi(s_3, W)$

Q(s1,E)	Q(s2,E)	Q(s3,S)	Q(s4,W)
0	0	0	10
0	0	8	10
0	<b>6.4</b>		



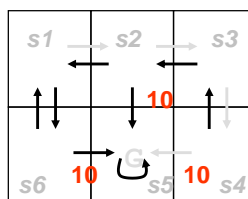
# Q-Learning Iterations - trial 3



- Start at upper left – move clockwise;  $\gamma = 0.8$

$$Q_{k+1}^\pi(s_3, S) = Q_k^\pi(s_3, S) + \alpha [r' + \gamma \max_{a'} Q_k^\pi(s_4, a') - Q_k^\pi(s_3, S)]$$

$$Q_{k+1}^\pi(s_3, S) = 0 + 1 [0 + 0.8 \{ \max, 10, 0 \} - 0] = 8$$



$Q_k^\pi(s_4, W)$        $Q_k^\pi(s_4, N)$

Q(s1,E)	Q(s2,E)	Q(s3,S)	Q(s4,W)
0	0	0	10
0	0	8	10
0	<b>6.4</b>	8	10



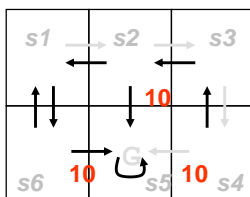
# Q-Learning Iterations - trial 4



- Start at upper left – move clockwise;  $\gamma = 0.8$

$$Q_{k+1}^\pi(s_1, E) = Q_k^\pi(s_1, E) + \alpha [r' + \gamma \max_{a'} Q_k^\pi(s_2, a') - Q_k^\pi(s_1, E)]$$

$$Q_{k+1}^\pi(s_1, E) = 0 + 1 [0 + 0.8 \times \max_{a'} \{6.4, 0, 0\} - 0] = 5.12$$



Q(s1,E)	Q(s2,E)	Q(s3,S)	Q(s4,W)
0	0	0	10
0	0	8	10
0	6.4	8	10
<b>5.12</b>			

Potrei migliorare la policy: dovrei scegliere South in  $s_2$

<http://borghese.di.unimi.it/>





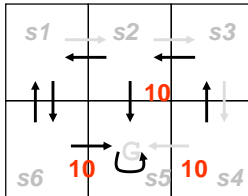
## Q-Learning Iterations: improving policy



- Start at upper left – move clockwise;  $\gamma = 0.8$ ;  $\alpha = 1$

$$Q_{k+1}^{\pi}(s_2, S) = Q_k^{\pi}(s_2, S) + \alpha [r' + \gamma \max_{a'} Q_k^{\pi}(s_2, a') - Q_k^{\pi}(s_2, S)]$$

$$Q_{k+1}^{\pi}(s_2, S) = 0 + 1[10 + 0.8 \times 0 - 0] = 8$$



$Q_k^{\pi}(s_5, \cdot)$

Mossa  $\epsilon$ -greedy in  $s_2$  (invece che  $a = E$ , scelgo  $a = S$ , cambio azione):  
calcolo  $Q(s_2, S) = r + \gamma \max_{a'} \{Q(s_2, a')\} = 10 + 0.8 \times 0 = 10$

Q(s1,E)	Q(s2,E)	Q(s2,S)	Q(s3,S)	Q(s4,W)
0	0		0	10
0	0		8	10
0	6.4		8	10
5.12	6.4	8	8	10

ni.it



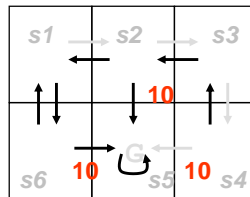
## Q-Learning Iterations: policy changed!



- Start at upper left – move clockwise;  $\gamma = 0.8$

$$Q_{k+1}^{\pi}(s_1, E) = Q_k^{\pi}(s_1, E) + \alpha [r' + \gamma \max_{a'} Q_k^{\pi}(s_1, a') - Q_k^{\pi}(s_1, E)]$$

$$Q_{k+1}^{\pi}(s_1, E) = 0 + 1 [0 + 0.8 \times \max_{a'} \{6.4, 8, 0\} - 0] = 6.4$$



$Q_k^{\pi}(s_2, E)$

$Q_k^{\pi}(s_2, W)$

$Q_k^{\pi}(s_2, S)$

Q(s1,E)	Q(s2,E)	Q(s2,S)	Q(s3,S)	Q(s4,W)
0	0		0	10
0	0		8	10
0	6.4		8	10
6.4	6.4	8	8	10

ni.it



# Proprietà del rinforzo

L'ambiente o l'interazione può essere complessa.

Il rinforzo può avvenire solo dopo una più o meno lunga sequenza di azioni (**delayed reward**).

E.g. agente = giocatore di scacchi.  
ambiente = avversario.

**Problemi collegati:**  
temporal credit assignment.  
structural credit assignment.

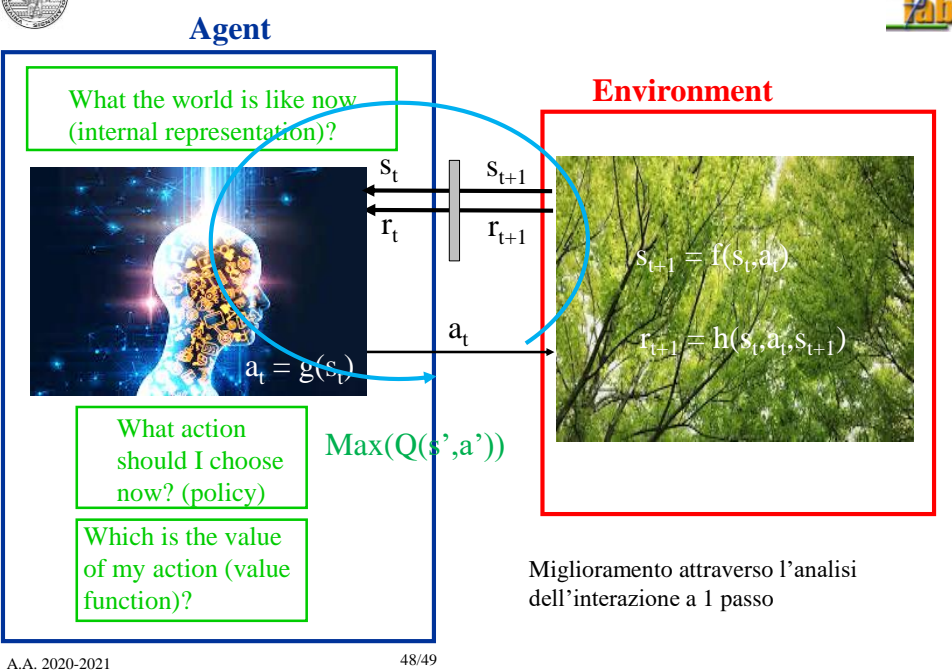
L'apprendimento non è più da esempi, ma dall'osservazione del proprio comportamento nell'ambiente.

Utilizzo delle equazioni di Bellman

Utilizzo una "porzione" (sample) di esperienza per migliorare la policy.



# Un ciclo di interazione







# Sommario



SARSA

Q-learning

Esempi