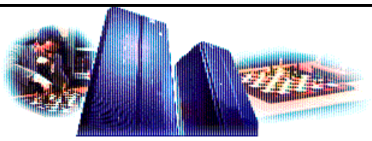
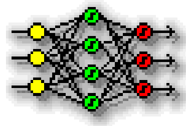




Sistemi Intelligenti Reinforcement Learning: Q-learning

Alberto Borghese

Università degli Studi di Milano
Laboratorio di Sistemi Intelligenti Applicati (AIS-Lab)
Dipartimento di Informatica
alberto.borghese@unimi.it

Barto and Sutton, 4.7, 6.4, 6.5

Sommario

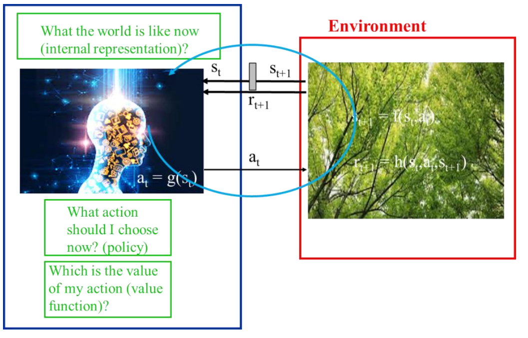
Q-learning

Esempi

Value Function?

La Value Function deriva dalla visione della Programmazione Dinamica e dell'ottimizzazione.

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



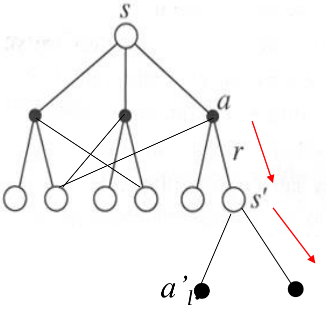
La policy in SARSA

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

1) Apprendiamo il valore di Q^{π} (s,a), ∀s ∀a, per una policy data (***on-policy***).

2) Dopo avere appreso la funzione Q^{π} (s,a), possiamo **modificare la policy, π'(s,a),** in modo da migliorarla (**policy improvement**)

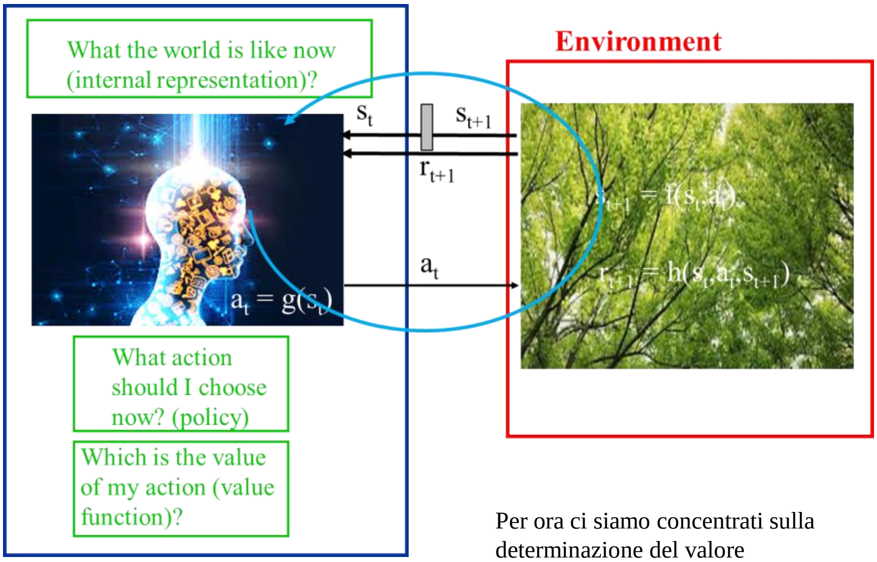
3) Dopo avere modificato la policy devo apprendere la nuova $Q^{\pi'}$(s,a)



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

On-policy learning.

Un ciclo di interazione



Per ora ci siamo concentrati sulla determinazione del valore

Value iteration

Iterative policy evaluation

$$Q_{k+1}^{\pi}(s_t, a_t) = \sum_{s'} \mathrm{P}_{s \to s' | a} \left\{ R_{s,s',a} + \gamma\, \pi(s', a') \sum_{a'} Q_k^{\pi}(s'_{t+1}, a'_{t+1}) \right\}$$

Invece di considerare una policy stocastica, consideriamo l'azione migliore:

$$Q_{k+1}(s_t, a_t) = \max_{a'} \sum_{s'} \mathrm{P}_{s \to s' | a} \left\{ R_{s,s',a} + \gamma\, \pi(s', a') \sum_{a'} Q_k\, (s', a') \right\}$$

$\forall$**s**

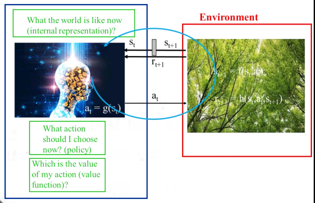
Off-policy Temporal Difference: Q-learning

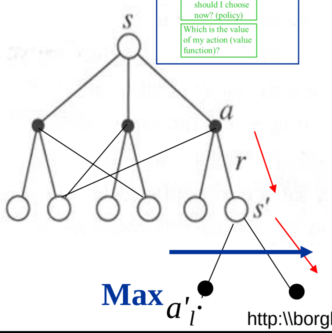
$$Q_{k+1}(s,a) = Q_k^{\pi}(s,a) + \alpha \left[r' + \gamma \max_{a'} Q_k\,(s',a') - Q_k\,(s,a) \right]$$

Non imparo semplicemente la funzione valore $Q^{\pi}(.)$, ma la funzione valore $Q^{*}(.)$ 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, $a=\pi(s)$ -> $a_{new}=\pi'(s)$ senza aspettare di avere stimato esattamente $Q^{\pi}(.)$.



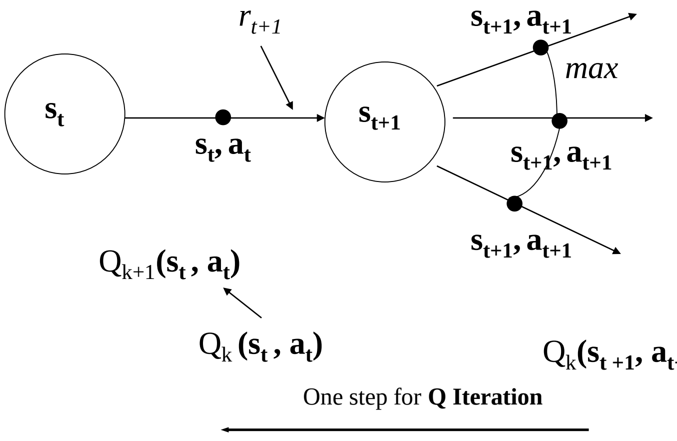


Q-learning algorithm (progetto)

```
Q(s,a) = 0;              // ∀s, ∀a,
Policy data;             // deterministica o stocastica
Repeat                                    // for each episode
{   s = s0; α = α*reduction_factor;       // decremento il coefficiente di aggiornamento α
    Repeat                                // for each step of the single episode
    {   a = Policy(s);                    // Policy
        s_next = NextState(s,a);          // non nota all'agente
        reward = Reward(s, s_next, a);            // non nota all'agente
        a_next_pol = Policy (s_next);             // Policy
        a_next = argmax(Q(s_next, a));            // Azione greedy
                    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) )
```

We may decide to quit when the policy does not change for several episodes.

Rappresentazione grafica



Viene migliorata la policy al tempo t+1 (off-policy)
Al tempo t, nello stato s_t, l'agente sceglie l'azione a_t. Arriva in s_{t+1} e "ragiona" su come continuare

Osservazioni

π(s,a): l'agente sceglie l'azione ottima

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

Nella pratica la convergenza viene valutata sulle variazioni di Q, ma anche sulla stabilità della policy identificata.

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

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

Q-learning è **off-policy** perchè la policy viene variata all'interno dell'algoritmo.

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.

Sw del labirinto

Sommario

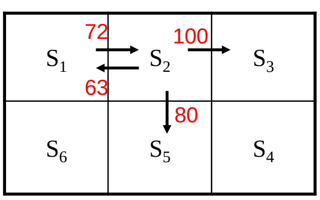
Q-learning

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 $\{s_1 \ldots s_6\}$
Azioni: {su, destra, giù, sinistra}
Reward istananteo = 0
Inizializzo Q(s,a) – in rosso.

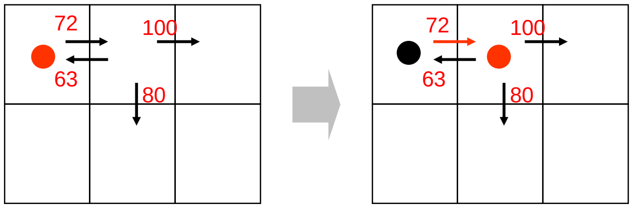


In rosso i valori di Q(s,a).
Nessun reward istantaneo.

Example 1 – Q Learning Update

$\gamma = 0.9$

$s_{ini} = s_1$



0 reward received in the transitions. Q(s,a) initialized ≠ 0

S_1	S_2	S_3
S_6	S_5	S_4

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

Apprendimento della funzione valore Q. Versione Q-learning.

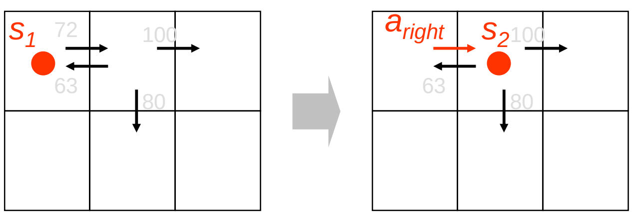
Iniziamo con una policy ben definita. Supponiamo: *right* = $\pi(s_1)$ -> $Q(s_1,dx)$ = ?

Example 1 – Q Learning: Policy Update

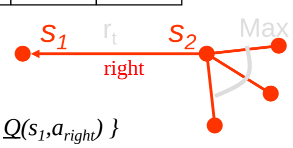
$\gamma = 0.9$

$\alpha = 0.1$

$a(s_2)$ = down



0 reward received in the transition



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

Correzione di $\underline{Q}(s_1,a_{right})$

Correzione dell'azione in s_2 da down a right

Viene modificata la policy da s_2 in poi

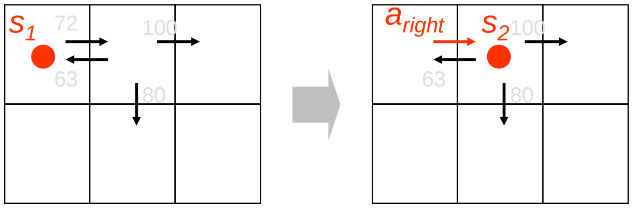
$Q(s_2,a_{down}) = 80$

$Q(s_2,a_{right}) = 100$

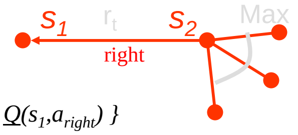
$Q(s_2,a_{left}) = 63$

Example 1 – Q Learning Update

$\gamma = 0.9$
$\alpha = 0.1$
$a(s_2)$ = down



0 reward received in the transition



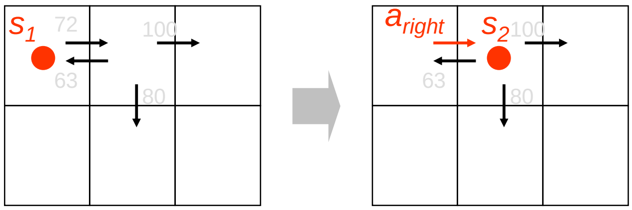
$Q(s_1,a_{right}) = Q(s_1,a_{right}) + \alpha\{ r(s_1,a_{right}, s_2) + \gamma \max_{a'} Q(s_2,a') - Q(s_1,a_{right}) \}$
$= 72 + \alpha \{0 + 0.9 \max_{a'} \{63, 80, 100\} - Q(s_1,a_{right}) \}$
$= 72 + \alpha (0+0.9*100 - 72) = 72 + 0,1*18 = 73.8$

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

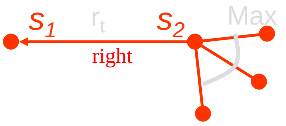
$Q(s_2,a_{down}) = 80$
$Q(s_2,a_{right}) = 100$
$Q(s_2,a_{left}) = 63$

Example 1 – Q Learning Update series

$\gamma = 0.9$
$\alpha = 0.1$
$a(s_2)$ = down



0 reward received in the transition

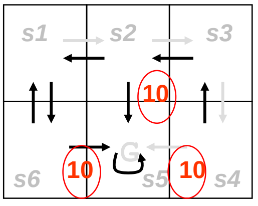


$Q(s_1,a_{right}) = 72 + \alpha (90 - 72) = 72 + 1.8 = 73.8$ trial 1
$Q(s_1,a_{right}) = 73.8 + \alpha (90 - 73.8) = 73.8 + 1.62 = 75.42$ trial 2
$Q(s_1,a_{right}) = 75.42 + \alpha (90 - 75.42) = 75.42 + 1.458 = 76.878$ trial 3
$Q(s_1,a_{right}) = 76.878 + \alpha (90 - 76.878) = 76.878 + 1.3122 = 78.1902$ trial 4
$Q(s_1,a_{right}) = 78.1902 + \alpha (90 - 78.1902) = 75.42 + 1.458 = 79.37118$ trial 5
$Q(s_1,a_{right}) = 79.37118 + \alpha (90 - 79.37118) = 75.42 + 1.458 = 80.434062$ trial 6
……
Si ottiene una serie che converge al valore asintotico 90 (asintoticamente)

Example 2: Q-Learning Iterations

- Stati: $\{s_1, \dots s_6\}$
- Azioni: {dx. sx, su, giù}
- **Reward istantaneo solo in alcune transizioni (in rosso e cerchiato)**.
- Q(s,a) = 0 per tutti gli stati.
- Stato iniziale: s_1

- Initial selected policy: move clockwise;



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

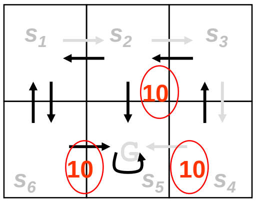
$\alpha = 1$
$\gamma = 0.8.$

Example 2: Q-Learning Iterations

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

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

Reward istanteo in rosso e cerchiato



$$Q_{k+1}^{\pi}(s_1,E) = 0 + 1[0 + 0.8x0 - 0] = 0$$

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

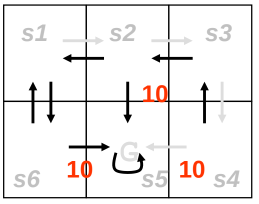
Q(s_1,East)	Q(s_2,East)	Q(s_3,South)	Q(s_4,West)
0			

Q-Learning Iterations – trial 1

- Start at upper left – move clockwise; table initially 0; γ = 0.8; α = 1

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

$$Q^{\pi}_{k+1}(s_3,S) = 0 + 1[0 + 0.8x0 - 0] = 0$$



Q(S1,E)	Q(s2,E)	Q(s3,S)	Q(s4,W)
0	0	0	

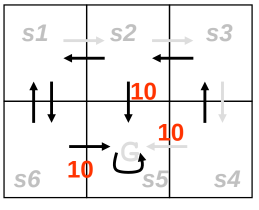
Q-Learning Iterations – trial 1

- Start at upper left – move clockwise; γ = 0.8

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

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

$Q^{\pi}_{k}(s_5,.)$ goal



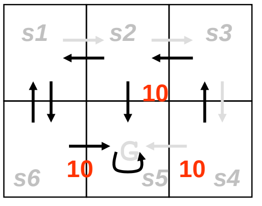
Q(S1,E)	Q(s2,E)	Q(s3,S)	Q(s4,W)
0	0	0	**10**

Q-Learning Iterations – trial 2

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

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

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



$Q^{\pi}_{k}(s_4, W)$ $Q^{\pi}_{k}(s_4, N)$

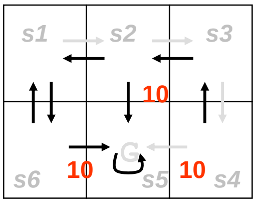
Q(S1,E)	Q(s2,E)	Q(s3,S)	Q(s4,W)
0	0	0	10
0	0	**8**	

Q-Learning Iterations – trial 2

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

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

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



$Q^{\pi}_{k}(s_5, .)$ goal

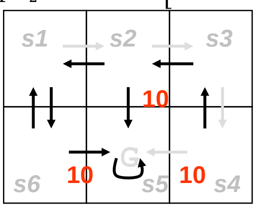
Q(S1,E)	Q(s2,E)	Q(s3,S)	Q(s4,W)
0	0	0	10
0	0	*8*	10

Q-Learning Iterations – trial 3

- Start at upper left – move clockwise; γ = 0.8

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

$$Q^{\pi}_{k+1}(s_2,E) = 0 + 1\left[0 + 0.8\ x \max_{a'}\{8\ ,0\} - 0\right] = 6.4$$



$Q^{\pi}_{k}(s_3,S)$ $Q^{\pi}_{k}(s_3,W)$

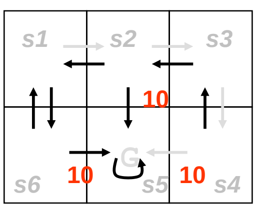
Q(S1,E)	Q(s2,E)	Q(s3,S)	Q(s4,W)
0	0	0	10
0	0	8	10
0	**6.4**		

Q-Learning Iterations – trial 3

- Start at upper left – move clockwise; γ = 0.8

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

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



$Q^{\pi}_{k}(s_4,N)$ $Q^{\pi}_{k}(s_4,W)$

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

Q-Learning Iterations – trial 4

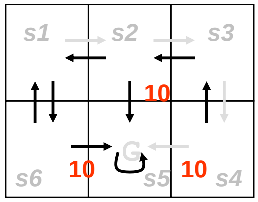
- Start at upper left – move clockwise; $\gamma = 0.8$

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

$$Q^{\pi}_{k+1}(s_1,E) = 0 + 1\left[0 + 0.8\,x\,\max_{a'}\{6.4, 0, 0\} - 0\right] = 5.12$$

$Q^{\pi}_{k}(s_2,W)$

$Q^{\pi}_{k}(s_2,E)$ $\quad Q^{\pi}_{k}(s_2,S)$



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
5.12	6.4	8	10

Potrei migliorare la policy: dovrei scegliere South in s_2

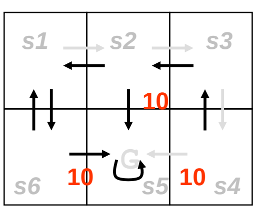
Q-Learning Iterations: improving policy

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

$$Q^{\pi}_{k+1}(s_2,S) = Q^{\pi}_{k}(s_2,S) + \alpha\left[r' + \gamma \max_{a'} Q^{\pi}_{k}(s_5,a') - Q^{\pi}_{k}(s_2,S)\right]$$

$$Q^{\pi}_{k+1}(s_2,S) = 0 + 1[10 + 0.8\,x\,0 - 0] = 10$$

$Q^{\pi}_{k}(s_5,.)$



Mossa ε-greedy in s_2 (invece che a = E, scelgo a = S, cambio azione):
calcolo Q(s_2, S) = $r + \gamma \max_{a'}$ {Q(s5,a')}= 10 + 0.8 x 0 = **10**

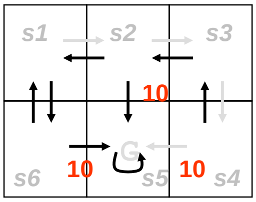
Q(S1,E)	Q(s2,E)	Q(s2,S)	Q(s3,S)	Q(s4,W)
0	0	0	0	10
0	0	0	8	10
0	6.4	0	8	10
5.12	6.4	10	8	10

Q-Learning Iterations: policy changed!

- Start at upper left – move clockwise; γ = 0.8

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

$$Q^{\pi}_{k+1}(s_1,E) = 5.12 + 1\left[0 + 0.8\ x \max_{a'}\{6.4, 10, 0\} - 5.12\right] = 8$$



$Q^{\pi}_{k}(s_2,W)$

$Q^{\pi}_{k}(s_2,E)$ $Q^{\pi}_{k}(s_2,S)$

Q(S1,E)	Q(s2,E)	Q(s2,S)	Q(s3,S)	Q(s4,W)
0	0	0	0	10
0	0	0	8	10
0	6.4	0	8	10
8	6.4	10	8	10

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.

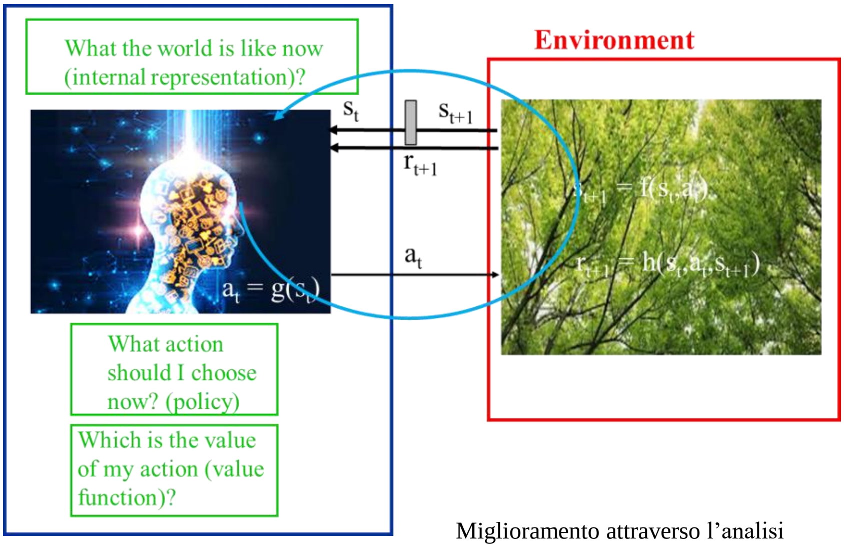
Problemi collegati:
temporal credit assignement.
structural credit assignement.

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



Miglioramento attraverso l'analisi dell'interazione a 1 passo

Sommario

Q-learning

Esempi