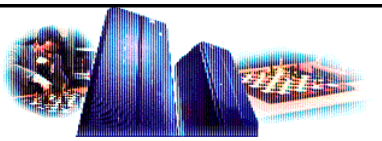
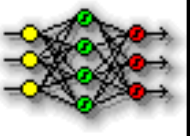


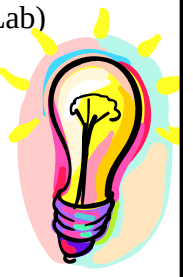


Sistemi Intelligenti
Reinforcement Learning: SARSA and Q-learning

Alberto Borghese

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

Barto and Sutton, 4.7, 6.4, 6.5



Sommario

SARSA

Q-learning

Esempi

Meccanismo di apprendimento nel RL

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

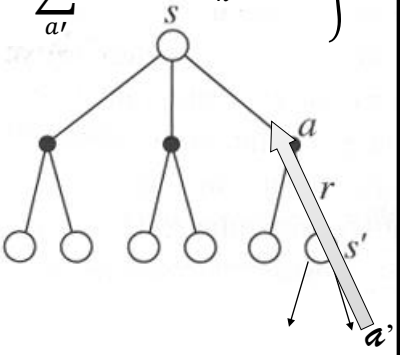
Ciclo dell'agente (le tre fasi sono sequenziali):

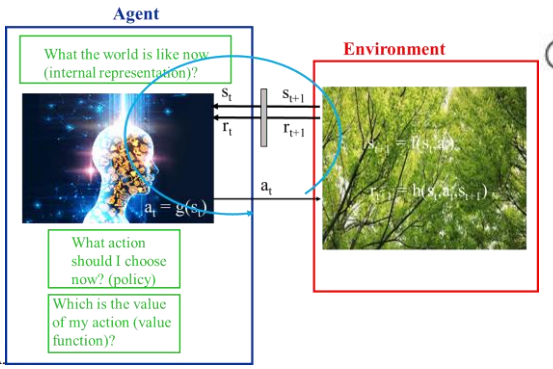
1) Implemento una policy (π(s,a))
2) Calcolo la Value function (Q^{π}(s,a))
3) Aggiorno la policy.

Un ciclo di interazione

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

Calcolo ricorsivo di Q(.)





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.

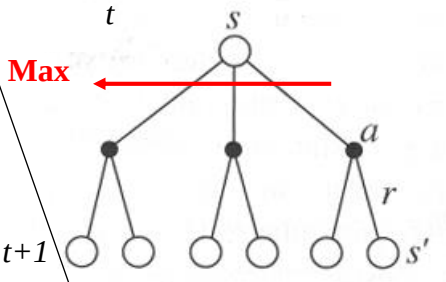
Ciclo dell'agente (le tre fasi sono sequenziali):

1) Implemento una policy (π(s,a))
2) Determino la Value function (Q^{π}(s,a))
3) Aggiorno la policy.

Q(s,a) – Osservazioni

$$Q^{\pi}(s_t, a_t) = \sum_{s'} \mathrm{P}_{s \to s'|a} \left\{ R_{s,s',a} + \gamma \sum_{a'} \pi(s', a') Q^{\pi}(s', a') \right\}$$

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:

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

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

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

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).

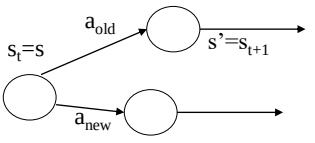
Teorema del miglioramento della policy

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

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

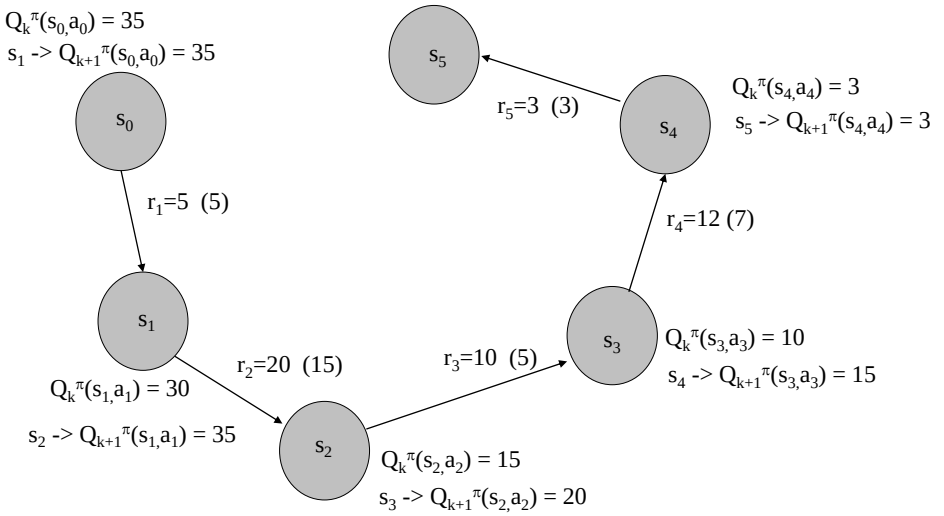
$Q^{\pi'}(s_t, a_{new} = \pi'(s_t)) \geq Q^{\pi}(s_t, a_t)$ $\forall a_t$

Tesi: π' è meglio di π. Cioè: $Q^{\pi'}(s, \pi'(s)) >= Q^{\pi}(s, \pi(s))$ ∀s.



Learning $Q^{\pi}(s,a)$

s_0 = ufficio; s_5 = casa.



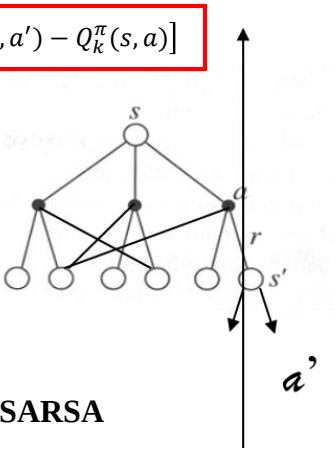
Come i diversi reward istantanei modificano $Q^{\pi}(s,a)$?

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 α decresce dolcemente a 0 all'aumentare del numero di trial

.

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



Sample backup, single state, s_t, single action, a_t, single future state $s'=s_{t+1}$

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

Meccanismo di apprendimento nel RL

Inizializzazione: se l'agente non agisce sull'ambiente non succede nulla. Occorre specifiare 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.

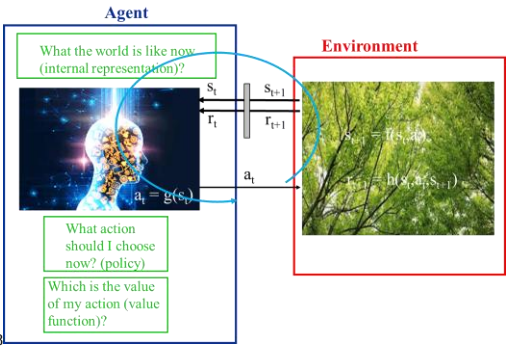
Background su SARSA

Al tempo t abbiamo a disposizione:

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

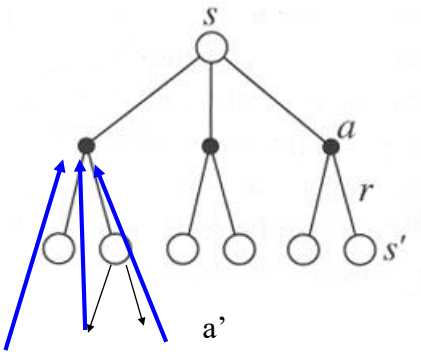
$s_{t+1} = s'$ estratto (sampled) dalla distribuzione statistica: $P_{s \to s'|a_j}$

Dopo la realizzazione di un evento, l'incertezza statistica scompare.
1 Reward certo
1 Transizione certa
vengono forniti dall'ambiente

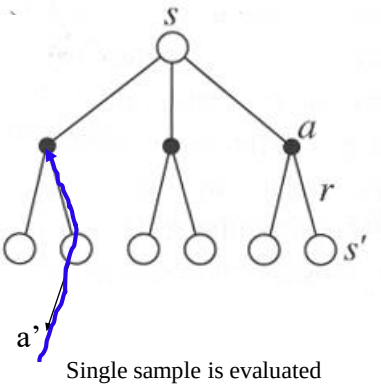


Sample backup

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



Full backup



Single sample is evaluated

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

SARSA Algorithm (progetto)

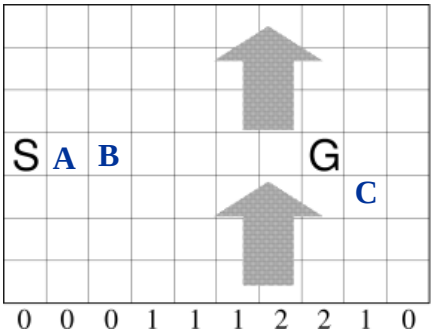
```
Q(s,a) = rand();            // ∀s, ∀a, eventualmente Q(s,a) = 0 - inizializzazione
Policy definita;            // Policy specificata, eventualmente stocastica
Repeat                      // for each episode
{  s = s0;
   Repeat                   // for each step of the actual episode
   {
        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?

Esempio

From Start, S, to Goal, G.

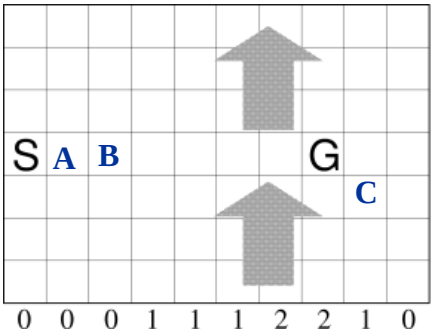


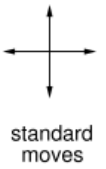
Stati = {caselle della griglia}
Stato iniziale = S
Terminal state = G
Azioni = {su, destra, giù, sinistra}
Reward = -1 tranne che quando s' = TS (reward = 0)

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

Esempio

From Start, S, to Goal, G.





Upwards wind

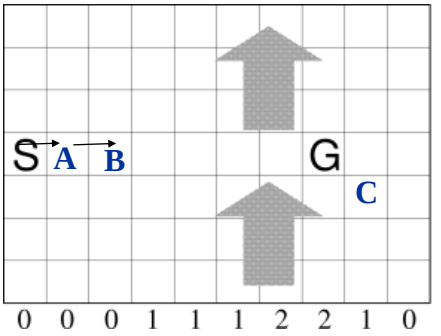
Q(s,a) iniziale = 0.
r = 0 se s' = G; altrimenti r = -1.
π(s,a) data.

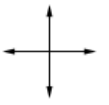
$\alpha = 0.5$
$\gamma = 1$

Vogliamo valutare π(s,a).

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

Esempio - risultato



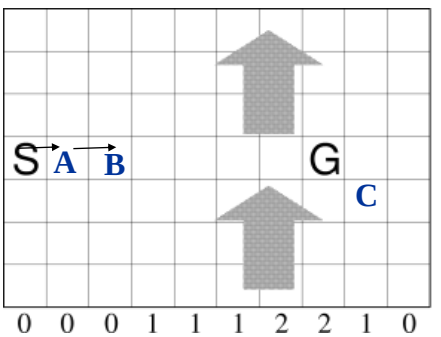


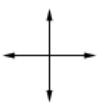
ε = 0.1
α= 0.5
γ = 1

Correzione di Q ad un passo:
$Q^{\pi}_{k+1}(S, east) = Q^{\pi}_{k}(S, east) + \alpha\left[r' + \gamma Q^{\pi}_{k}(A, east) - Q^{\pi}_{k}(S, east)\right]$= 0 + 0.5 [-1+1x0–0] = -0.5

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

Esempio - risultato





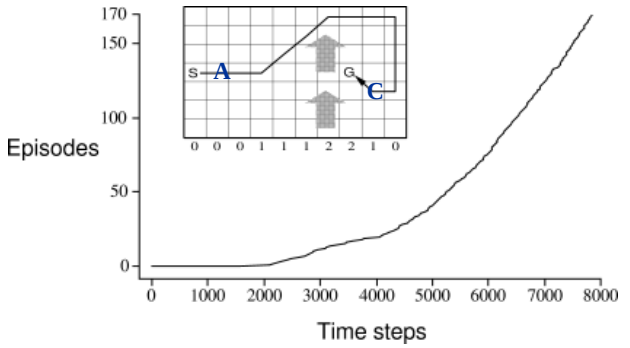
α= 0.5
γ = 1

Correzione di Q ad un passo:
$Q^{\pi}_{k+1}(A, east) = Q^{\pi}_{k}(A, east) + \alpha\left[r' + \gamma Q^{\pi}_{k}(B, east) - Q^{\pi}_{k}(A, east)\right]$= 0 + 0.5 [-1+1x0–0] = -0.5

$Q^{\pi}_{k+1}(C, west) = Q^{\pi}_{k}(C, west) + \alpha\left[r' + \gamma Q^{\pi}_{k}(G, .) - Q^{\pi}_{k}(C, west)\right]$= 0 + 0.5 [0+1x0–0] = 0
(NB c'è il vento verso l'alto di 1)

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

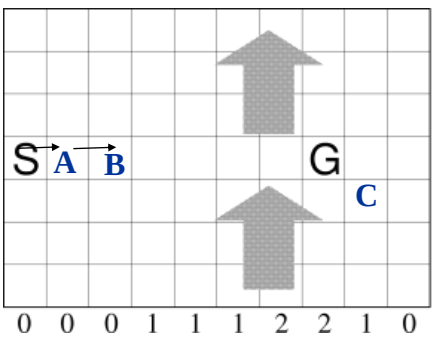
Esempio - risultato



Policy ε-greedy
ε = 0.1
α= 0.5
γ = 1

Aumentano gli episodi nello stesso intervallo di tempo, via via che trial vengono eseguiti.
All'inzio un trial richiede molto tempo per essere eseguito.
Non è il percorso ottimo (17 passi contro 15 passi)
E' il percorso cristallizzato.

Esempio - risultato



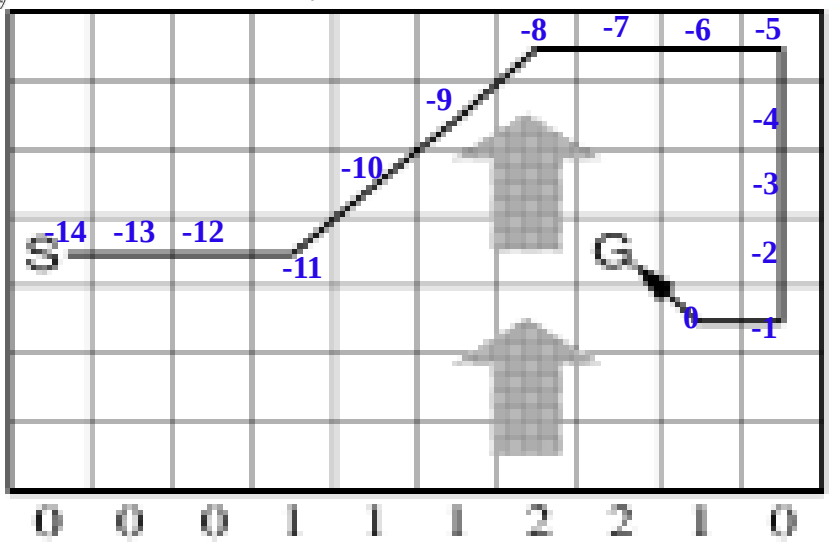
ε = 0.1
α= 0.5
γ = 1

Correzione di Q ad un passo:
$Q^{\pi}_{k+2}(S, east) = Q^{\pi}_{k+1}(S, east) + \alpha\left[r' + \gamma Q^{\pi}_{k+1}(A, east) - Q^{\pi}_{k+1}(S, east)\right] =$
-0.5 + 0.5 [-1+1x0 – 0.5] = -0.75

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

Esempio – valore di $Q^{\pi}(.)$

α= 1 γ = 1



We have estimated $Q^{\pi}(s,a)$. In most states we have better option. E.g. in S if we have better $Q^{\pi}(S,anew)$ than moving East. This allows exploring different paths. We can change after having estimated $Q^{\pi}(.)$.

Sommario

SARSA

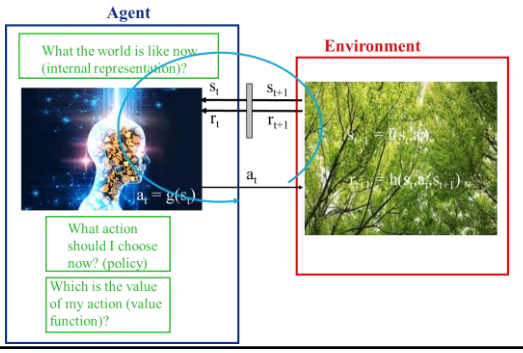
Q-learning

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.



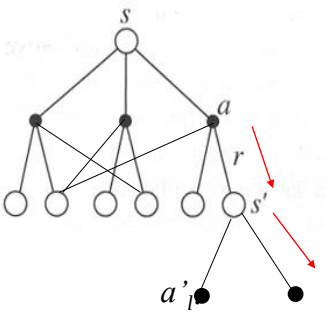
La policy in SARSA

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

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

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

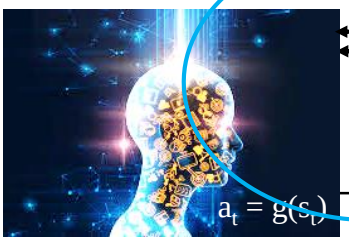
3) Dopo avere modificato la policy devo apprendere la nuova $Q^{\pi'}(.)$



Un ciclo di interazione

Agent

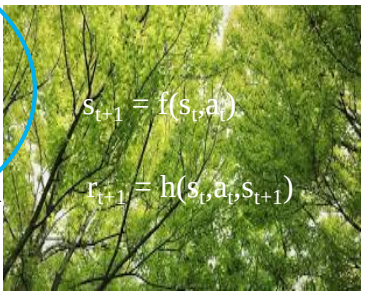
What the world is like now (internal representation)?



What action should I choose now? (policy)

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

Environment



s_t s_{t+1} r_t r_{t+1} a_t $a_t = g(s_t)$

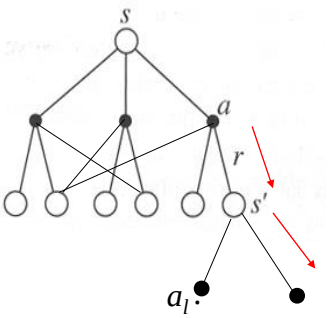
Per ora ci siamo concentrati sulla determinazione del valore

Come apprendere Q: SARSA

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

1) Apprendiamo il valore di Q per una policy data, π, (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.

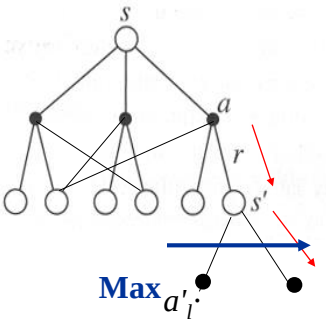
Off-policy Temporal Difference: Q-learning

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

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

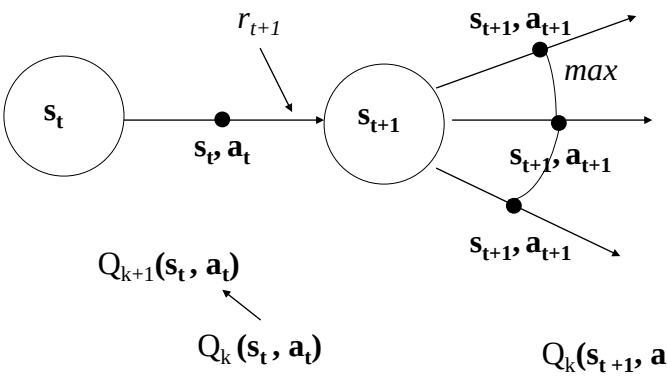
Eventualmente cambio subito policy, $a=\pi(s)$ -> $a_{new}=\pi'(s)$ senza aspettare di avere stimato esattamente $Q^{\pi}(.)$.



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);                           // 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) )
```

Rappresentazione grafica



One step for **Q Iteration**

Viene migliorata la policy al tempo t+1 (off-policy)

Osservazioni

π(s,a) sceglie l'azione ottima

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

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

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

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.

Sommario

SARSA

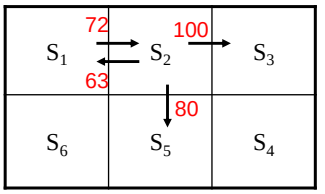
Q-learning

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

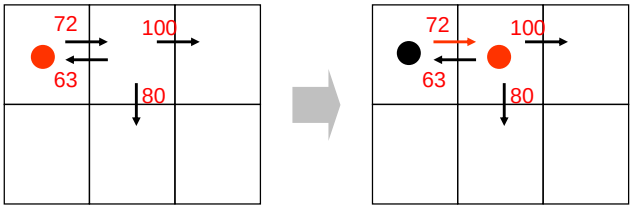


In rosso i valori di Q(s,a).
Nessun reward istantaneo.

Example 1 – Q Learning Update

γ = 0.9

$s_{ini} = s_1$



0 reward received in the transition. Q(.,.) initialized ≠ 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

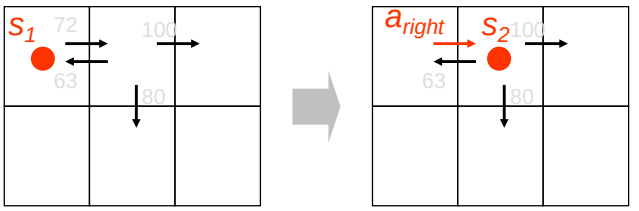
In rosso i valori di Q(s,a).
Nessun reward istantaneo.

Example 1 – Q Learning Update

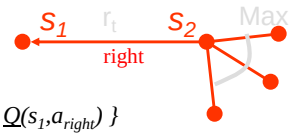
γ = 0.9

α= 0.1

$a(S_2)$ = down



0 reward received in the transition



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

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

$Q(s_1,a_{right}) = 90$

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

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

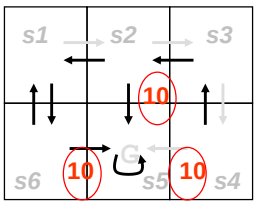
$Q(s_2,a_{left}) = 63$

Example 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



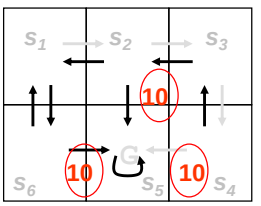
$\alpha = 1$
$\gamma = 0.8.$

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.

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

Reward istanteo in rosso e cerchiato



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

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

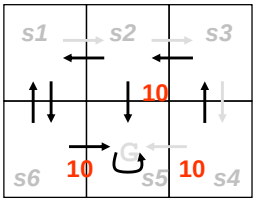
Q(s_1,East)	Q(s_2,East)	Q(s_3,South)	Q(s_4,West)
0			

Q-Learning Iterations – trial 1

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

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

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



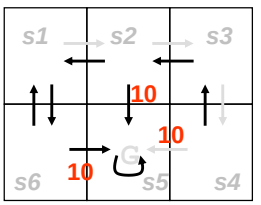
Q(S1,E)	Q(s2,E)	Q(s3,S)	Q(s4,W)
0	0	0	

Q-Learning Iterations – trial 1

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

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

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



$Q_k^{\pi}(s_5,.)$ goal

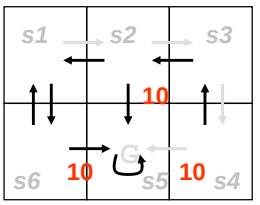
Q(S1,E)	Q(s2,E)	Q(s3,S)	Q(s4,W)
0	0	0	**10**

Q-Learning Iterations – trial 2

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

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

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



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

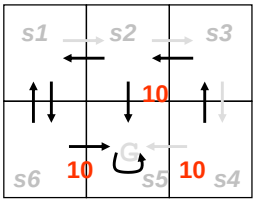
Q(S1,E)	Q(s2,E)	Q(s3,S)	Q(s4,W)
0	0	0	10
0	0	**8**	

Q-Learning Iterations – trial 2

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

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

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



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

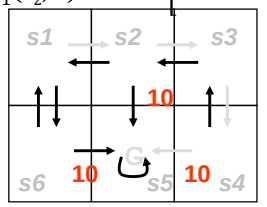
Q(S1,E)	Q(s2,E)	Q(s3,S)	Q(s4,W)
0	0	0	10
0	0	*8*	10

Q-Learning Iterations – trial 3

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

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

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



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

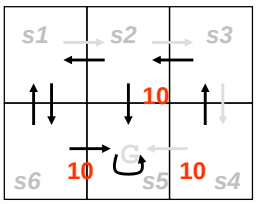
Q(S1,E)	Q(s2,E)	Q(s3,S)	Q(s4,W)
0	0	0	10
0	0	8	10
0	**6.4**		

Q-Learning Iterations – trial 3

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

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

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



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

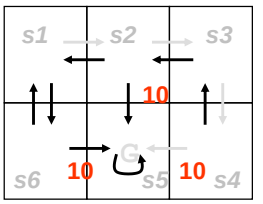
Q(S1,E)	Q(s2,E)	Q(s3,S)	Q(s4,W)
0	0	0	10
0	0	8	10
0	**6.4**	8	10

Q-Learning Iterations – trial 4

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

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

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



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

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

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

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

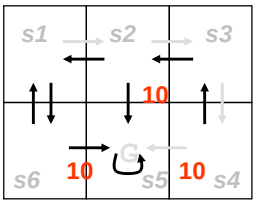
Potrei migliorare la policy: dovrei scegliere South in s_2

Q-Learning Iterations: improving policy

- Start at upper left – move clockwise; γ = 0.8; α = 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 $Q(s_2, S) = r + \gamma \max_{a'} \{Q(s5,a')\} = 10 + 0.8 \times 0 =$ **10**

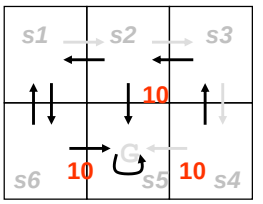
Q(S1,E)	Q(s2,E)	Q(s2,S)	Q(s3,S)	Q(s4,W)
0	0	0	0	10
0	0	0	8	10
0	6.4	0	8	10
5.12	6.4	10	8	10

Q-Learning Iterations: policy changed!

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

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

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



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

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

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

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.

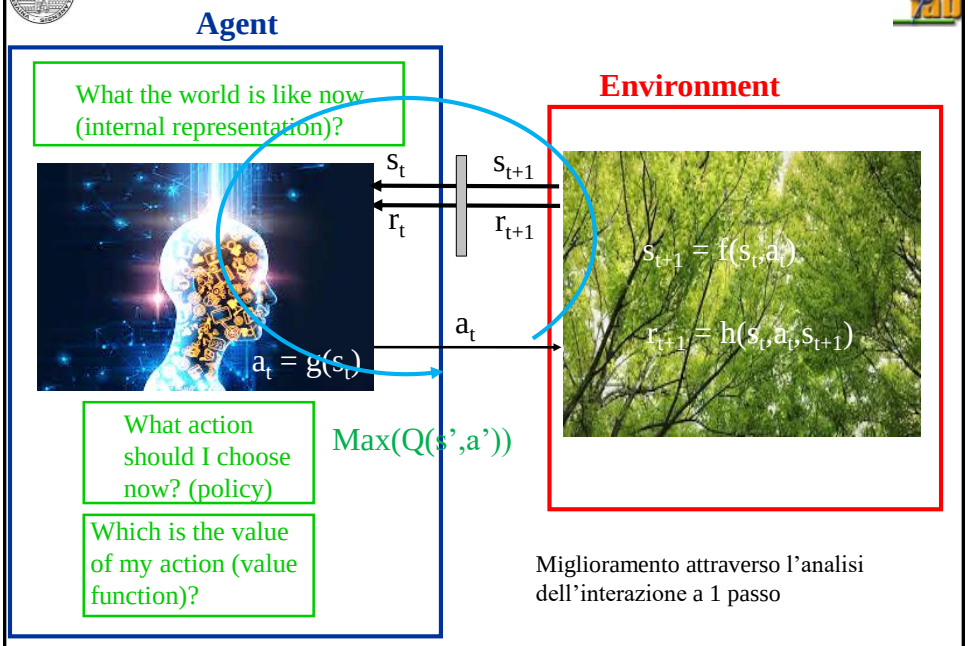
Problemi collegati:
temporal credit assignement.
structural credit assignement.

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

SARSA

Q-learning

Esempi