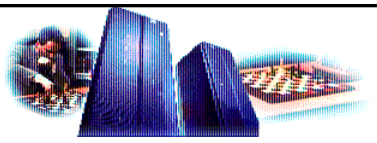
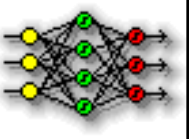




Sistemi Intelligenti
Reinforcement Learning: Temporal Differences

Alberto Borghese

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

Barto and Sutton, Capitoli 3 e 6

1

World RL competition

https://neurips.cc/Conferences/2024/CompetitionTrack
https://neurips.cc/virtual/2025/loc/san-diego/events/Competition

Started in NIP2006.

Many competitions, some are based on RL

Other competitions are organized

2

Sommario

Le equazioni di Bellman

Differenze temporali

SARSA

3

The RL updated picture

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)?

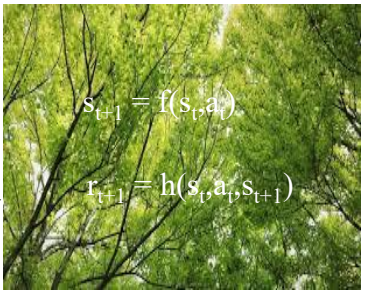
s_t

s_{t+1}

r_{t+1}

a_t

Environment



a_t, dipende dalla situazione!

4

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) **Aggiorno la Value function (Q^{π}(s,a))**
3) Aggiorno la policy.

Fino a quando la policy non è stabile.

Policy iteration.

5

Apprendimento

Iterative policy evaluation

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

Converge al limite a Q^{π}(s,a).

Policy improvement

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

Si arriva a:

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

6

Un ciclo di interazione

Agent

What the world is like now (internal representation)?

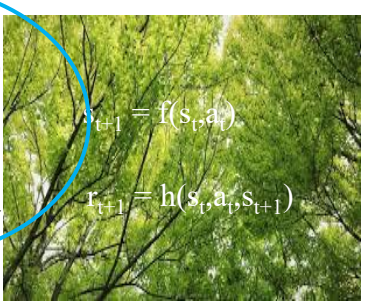
$a_t = g(s_t)$

What action should I choose now? (policy)

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

s_t s_{t+1} r_{t+1} a_t

Environment



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

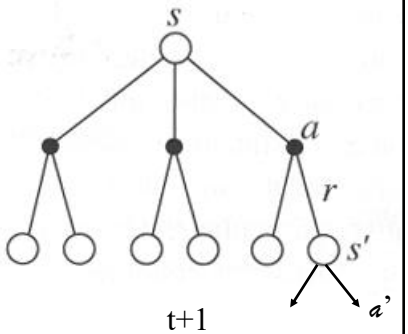
$r_{t+1} = h(s_t,a_t,s_{t+1})$

L'agente ragiona su quantità note a 1 passo

7

Tecnica full-back

Back-up



$\pi(s,a)$ fissata

t+1

Conosciamo $Q(s_t,a_t)$ $\forall s_t,a_t$ anche per $\{s'_{t+1}, a'_{t+1}\}$ quindi:

- Analizziamo la transizione da $\{s_t,a_t\}$ -> $\{s'_{t+1},a'_{t+1}\}$
- Calcoliamo un nuovo valore di Q per {s,a: $Q(s_t, a_t)$ congruente con:

  $Q(s_t, a_t)$ ed r_{t+1}

Full backup se esaminiamo tutti gli s' e a' (cf. DP).
Da {s', a'} mi guardo indietro e aggiorno Q(s,a).

π fissata

8

Meccanismo di apprendimento nel RL

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

1) Implemento una policy ($\pi(s,a)$)
2) Stimo la Value function ($Q^{\pi}(s,a)$) per tutte le coppie stato-azione
3) Miglioro la policy, $\pi(s,a)$.

Framework analizzati per calcolare $Q^{\pi}(s,a)$:

1. Stocastico completo. **L'agente conosce la statistica della dinamica dell'ambiente e dei reward**, scrive **le equazioni lineari** che legano le value function in stati diversi e **calcola** i valori di $Q^{\pi}(.)$.

2. Stocastico con aggiornamento. **L'agente conosce la statistica della dinamica dell'ambiente e dei reward.** Procede da uno stato iniziale a quello finale. **Da ogni stato esplora in parallelo tutti i possibili stati prossimi** e aggiorna i valori delle $Q^{\pi}(.)$. Iterative policy evaluation – Policy iteration.

3. Stocastico con interazione singola e con la scelta di una singola azione. **L'agente NON conosce la statistica della dinamica dell'ambiente e dei reward**. **Procede da uno stato iniziale a quello finale**. Da ogni stato esplora una sola azione e **un SOLO solo stato prossimo**. Aggiorna i valori di $Q^{\pi}(.)$.

9

Sommario

Le equazioni di Bellman

Differenze temporali

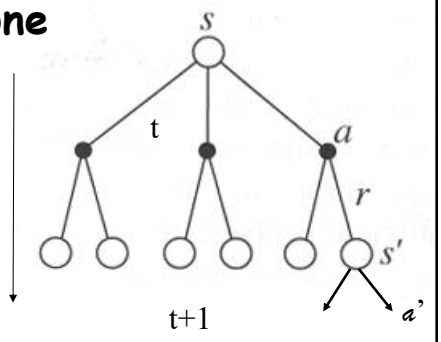
SARSA

10

1 passo di interazione

Back-up

π(s,a) fissata



Analizziamo la transizione da $\{s_t,a_t\}$ -> $\{s'_{t+1},a'_{t+1}\}$. Da {s', a'} mi guardo indietro e aggiorno Q(s,a).

- Nell'approccio iterative policy evaluation, considero TUTTI gli s'_{t+1} (tutti i reward a 1 passo con la probabilità di sceglierli, e i reward a lungo termine da s'_{t+1}: $Q(s'_{t+1},a'_{t+1})$) **∀s'**. *Full backup* se esaminiamo tutti gli s' e a' (cf. Dynamic Programming).

Come estendiamo al caso di un agente che esplora l'ambiente non noto? Non conosciamo tutti gli s'_{t+1} ma 1 SOLO s', s'_{t+1}, e misuro 1 SOLO reward a un passo; considero il reward TOTALE, a lungo termine, SOLO di 1 stato prossimo.

11

Q(s,a) – Osservazioni

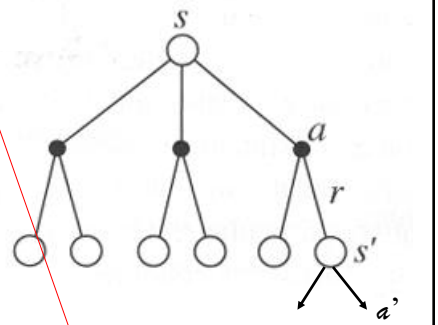
$$Q^{\pi}(s_t,a_t) = \sum_{s'} P_{s\to s'|a}\left\{R_{s,s',a} + \gamma \sum_{a'} \pi(s',a')Q^{\pi}(s',a')\right\}$$

Policy nota

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

$$a_{new} : \max_a Q(s,a)$$

Cosa cambia?



Non è noto il funzionamento dell'ambiente (interazione)

12

Background su Temporal Difference (TD) Learning

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

Come si possono utilizzare per apprendere **se non conosciamo l'ambiente**?

13

Confronto con il rinforzo classico

$$Q_{k+1} = Q_k - \frac{Q_k}{N_{k+1}} + \frac{r_{k+1}}{N_{k+1}} = Q_k + \alpha[r_{k+1} - Q_k]$$

Occupazione di memoria minima: Solo Q_k e k.
NB N_k è il numero di volte in cui è stata scelta a_j.

Questa forma è la base del RL. La sua forma generale è:

NewEstimate = OldEstimate + StepSize [Target – OldEstimate]
*NewEstimate = OldEstimate + StepSize * Error.*

StepSize = α = 1/(N+1) *a = cost*
Rewards weight w = 1 *Weight of i-th reward at time k: w = (1-a)$^{k-i}$*

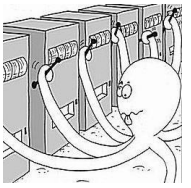
Qual è la differenza introdotta dall'approccio che prevede comportamenti (catene di azioni) e la valutazione di un reward TOTALE, a lungo termine?

14

Un possibile aggiornamento di Q(s,a)

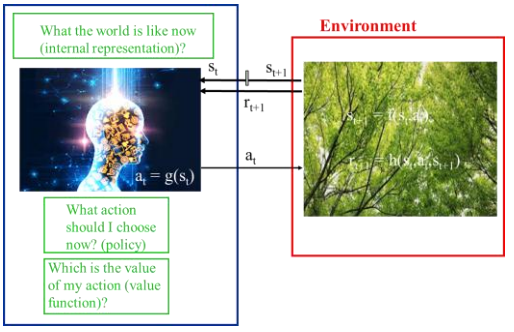
$$Q_{k+1}(a) = Q_k(a) - \frac{Q_k(a)}{N_{k+1}(a)} + \frac{r_{k+1}(a)}{N_{k+1}(a)} = Q_k(a) + \alpha[r_{k+1}(a) - Q_k(a)] =$$

$$Q_{k+1}(a) = Q_k(a) + \alpha\Delta Q_k(a) \qquad \alpha = \frac{1}{N_{k+1}(a)}$$



Come passo ai comportamenti?

$$Q^{\pi}_{k+1}(s,a) = Q^{\pi}_{k}(s,a) + \alpha\Delta Q_k(s,a)$$

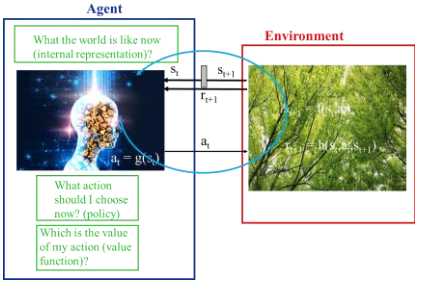


Come calcolo $\Delta Q_k(s,a)$?

Calcolo di $\Delta Q_k(s,a)$

$$Q_{k+1}(a) = Q_k(a) + \alpha[r_{k+1}(a) - Q_k(a)] =$$

$$Q_{k+1}(a) = Q_k(a) + \alpha\Delta Q_k(a)$$



Al tempo t abbiamo a disposizione:

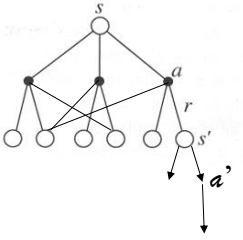
$r_{t+1} = r'$ da: $R_{s\to s'|a_j}$

$s_{t+1} = s'$ da: $P_{s\to s'|a_j}$

Quale semantica hanno Q(s,a) e r(s,a,s') nel caso dei comportamenti?

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

$$Q_{k+1}(s,a) = Q_k(s,a) + \alpha\Delta Q_k(s,a)$$
$$Q_{k+1}(a) = Q_k(a) + \alpha\Delta Q_k(a)$$

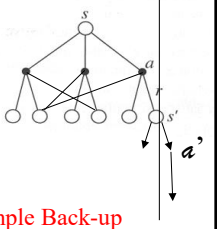


Reward a 1 passo Reward a lungo termine da s'

TD(0) update

Ad ogni istante di tempo di ogni trial aggiorno la Value function:

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



Sample Back-up

Conosciamo $Q(s_t,a_t)$ $\forall s_t,a_t$ anche per $\{s'_{t+1}, a'_{t+1}\}$ quindi:

- Analizziamo la transizione da $\{s_t,a_t\}$ -> $\{s'_{t+1},a'_{t+1}\}$
- Calcoliamo un nuovo valore di Q per {s,a}: $Q(s_t, a_t)$ congruente con: $Q(s_t, a_t)$ ed r_{t+1}

Sample backup se esaminiamo una sola coppia di s' e a' (cf. DP asincrona).
Da {s', a'} mi guardo indietro e aggiorno Q(s,a).
Percorro un solo ramo dell'albero, alla volta.

Per α che diminuisce con l'apprendimento, per k → ∞, idealmente come 1/(k+1), $Q^{\pi}_{k}(s,a)$ converge al valore vero di $Q^{\pi}(s,a)$

$\pi(s,a)$ fissata

Posso ragionare a un passo per calcolare $Q^{\pi}(s,a)$

17

Confronto con il setting associativo

$$Q_{k+1} = Q_k - \frac{Q_k}{N_{k+1}} + \frac{r_{k+1}}{N_{k+1}} = Q_k + \alpha[r_{k+1} - Q_k]$$

Occupazione di memoria minima: Solo Q_k e k.
NB k è il numero di volte in cui è stata scelta a_j.

Questa forma è la base del RL. La sua forma generale è:

NewEstimate = OldEstimate + StepSize [Target – OldEstimate]
*NewEstimate = OldEstimate + StepSize * Error.*

$StepSize = \alpha = 1/N_{k+1}$ $\quad a = cost$

18

Setting α value

$\alpha(s_t, a_t, s_{t+1}) = \frac{1}{N(s_t, a_t, s_{t+1})}$, where $N(s_t, a_t, s_{t+1})$ represents the number of occurrences of s_t, a_t, s_{t+1}. With this setting the estimated Q tends to the expected value of Q(s,a).

Per semplicità si assume solitamente $\alpha < 1$ costante. In questo caso, Q(s,a) assume il valore di una media pesata dei reward a lungo termine collezionati a partire da (s,a), con peso: $(1-\alpha)^k$: *exponential recency-weighted average*.

α che decresce dolcemente a zero consente la convergenza del Sistema stocastico.

19

Esempio

Stima del tempo di percorrenza da casa all'ufficio su un percorso ben definite (policy deterministica).

La durata dei diversi segmenti può variare da giorno a giorno e quindi la stima della durata totale viene corretta conseguentemente.

La stima corrente del tempo totale è data dalla somma dei tempi per:

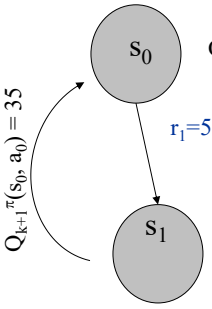
- Dall'ufficio (time to go = 35 minuti) - partito
- Dall'ufficio all'uscita del parcheggio: 5 minuti (time to go = 30 minuti)
- Dal parcheggio all'uscita dell'autostrada: 15 minuti (time to go = 15 minuti)
- Dall'uscita dell'autostrada alla strada di casa: 5 minuti (time to go = 10 minuti)
- Dalla strada di casa al parcheggio di casa: 7 minuti (time to go = 3 minuti)
- Dal parcheggio a casa: 3 minuti (time to go = 0 minuti) - arrivato

- In totale 35 minuti.

20

Learning $Q^{\pi}(s_0, a_0)$ - I

s_0 = ufficio; $Q_k^{\pi}(s_0, \text{vado\_parcheggio}) = 35\ minuti$ (potrei fare altre scelte, e.g. andare alla metropolitana, ma la policy prescrive di andare a prendere l'auto nel parcheggio perchè era considerata la soluzione più veloce).



$Q_k^{\pi}(s_0, \text{vado\_parcheggio}) = 35$ -> 35

Suppongo $\alpha = \gamma = 1$

r_1=5 minuti per uscire dal parcheggio

$Q_k^{\pi}(s_1, \text{imbocco\_autostrada}) = 30$

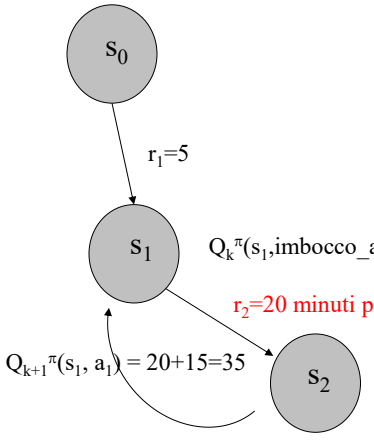
Raggiungo il parcheggio (stato s_1). Vedo che piove e spero che il tempo totale non vari di molto. Stimo il tempo per arrivare a casa in 30 minuti: all'uscita del parcheggio imbocco l'autostrada. Q_k^{π}(s_1. imbocco l'autostrada)

Ricalcolo il tempo totale, ovverosia il tempo dallo stato s_0 che non varia:

$$Q_{k+1}^{\pi}(s_0, a) = Q_k^{\pi}(s_0, a) + \alpha[r' + \gamma Q_k^{\pi}(s_1, a') - Q_k^{\pi}(s_0, a)] = 35+[5+30-35] = 35$$

Learning $Q^{\pi}(s_1, a_1)$ - II

s_1 = parcheggio; $Q_k^{\pi}(s_1, \text{imbocco\_autostrada}) = 30\ minuti$;
(potrei fare altre scelte, e.g. tornare in ufficio; una volta scelto di uscire, aggiorno il valore dell'azione uscire dal parcheggio, quando sono nel parcheggio)



Esco dall'autostrada (stato s_2). Sull'autostrada c'era traffico più lento del solito, impiego 20 minuti, 5 minuti in più del solito.

$Q_k^{\pi}(s_1, \text{imbocco\_autostrada}) = 30$ -> $Q_{k+1}^{\pi}(s_1, \text{imbocco\_autostrada}) = 35$

r_2=20 minuti per percorrere l'autostrada (solitamente erano 15 minuti)

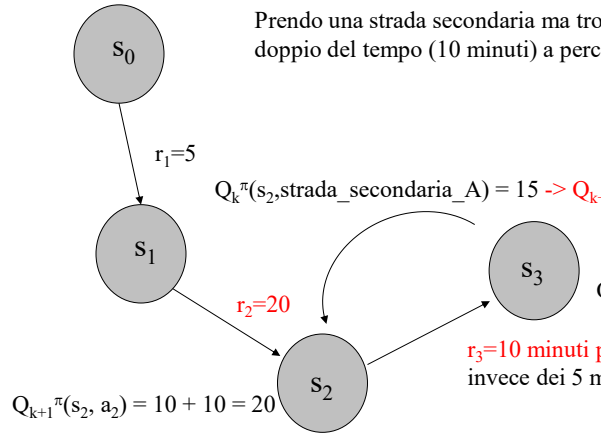
$Q_k^{\pi}(s_2, \text{strada\_secondaria\_A}) = 15$

Aggiorno il tempo totale dallo stato s_1:

$$Q_{k+1}^{\pi}(s_1, a) = Q_k^{\pi}(s_1, a) + \alpha[r' + \gamma Q_k^{\pi}(s_2, a') - Q_k^{\pi}(s_1, a)] = 30+[20+15-30] = 35$$

Learning $Q^{\pi}(s_2, a_2)$ - III

s_2 = esco_autostrada; $Q_k^{\pi}(s_2, \text{esco\_autostrada} = 15\ min;\ Q_{k+1}^{\pi}(s_2, \text{esco\_autostrada}) = 20\ min;$



Prendo una strada secondaria ma trovo dei lavori in corso… Ci metto il doppio del tempo (10 minuti) a percorrere la strada secondaria.

$Q_k^{\pi}(s_2,\text{strada\_secondaria\_A}) = 15$ -> $Q_{k+1}^{\pi}(s_2,\text{strada\_secondaria}) = 20$

$Q_k^{\pi}(s_3, \text{imbocco\_strada casa}) = 10$

r_3=10 minuti per percorrere la strada secondaria invece dei 5 minuti che ci mettevo di solito

$Q_{k+1}^{\pi}(s_2, a_2) = 10 + 10 = 20$

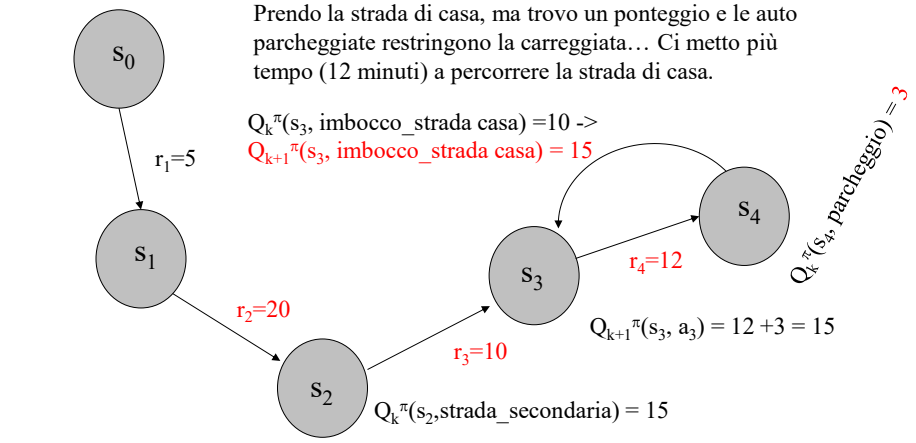
Aggiorno il tempo totale dallo stato s_2:

$$Q_{k+1}^{\pi}(s_2, a) = Q_k^{\pi}(s_2, a) + \alpha[r' + \gamma Q_k^{\pi}(s_3, a') - Q_k^{\pi}(s_2, a)] = 15+[10+10-15] = 20$$

23

Learning $Q^{\pi}(s_3, a_3)$ - IV

s_3 = imbocco_strada casa;$Q_k^{\pi}(s_3, \text{imbocco\_strada casa}) = 10\ min;$
$Q_{k+1}^{\pi}(s_3, \text{imbocco\_strada casa}) = 15\ min;$



Prendo la strada di casa, ma trovo un ponteggio e le auto parcheggiate restringono la carreggiata… Ci metto più tempo (12 minuti) a percorrere la strada di casa.

$Q_k^{\pi}(s_3, \text{imbocco\_strada casa}) = 10$ ->
$Q_{k+1}^{\pi}(s_3, \text{imbocco\_strada casa}) = 15$

$Q_{k+1}^{\pi}(s_3, a_3) = 12 + 3 = 15$

$Q_k^{\pi}(s_2,\text{strada\_secondaria}) = 15$

Aggiorno il tempo totale dallo stato s_2:

$$Q_{k+1}^{\pi}(s_2, a) = Q_k^{\pi}(s_3, a) + \alpha[r' + \gamma Q_k^{\pi}(s_3, a') - Q_k^{\pi}(s_3, a)] = 10+[12+3-10] = 15$$

24

Esempio – dopo il primo trial

Stima del tempo di percorrenza da casa all'ufficio su un percorso ben definite (policy deterministica).

La durata dei diversi segmenti può variare da giorno a giorno e quindi la stima della durata totale è stata corretta conseguentemente all'esplorazione.

La stima corrente del tempo totale è data dalla somma dei tempi per:

- Dall'ufficio all'uscita del parcheggio: 5 minuti (time to go = 35 minuti) - incongruenza
- Dal parcheggio all'uscita dell'autostrada: 15 minuti (time to go = 35 minuti)
- Dall'uscita dell'autostrada alla strada di casa: 5 minuti (time to go = 20 minuti)
- **Dalla strada di casa a casa: 7 minuti** **(time to go = 15 minuti)**
- **Dal parcheggio a casa: 3 minuti** **(time to go = 3 minuti)**

Si sono create diverse incongruenze (ad esempio il time to go è di 35 minuti dall'ufficio come dal parcheggio!), che verranno corrette via via che si ripeteranno le stesse situazioni.

Attualmente la stima aggiornata di Q(.) è per lo stato prima di quello finale ed è di 15 minuti. La stima di Q(.) per gli stati precedenti, viene via via aggiornata nei trial successivi.

In totale ci metto 50 minuti. Come i diversi reward istantanei modificano $Q^{\pi}(s,a)$ per tutti gli stati?

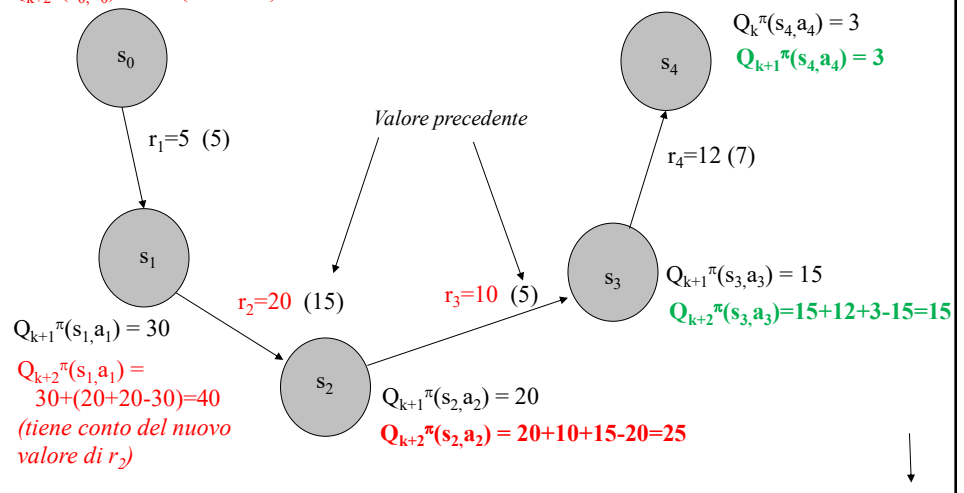
25

Learning $Q^{\pi}(s,a)$ – Trial 2

s_0 = ufficio; s_5 = casa.

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

$Q_{k+2}^{\pi}(s_0,a_0) = 35+(5+35-35)=40$ - corretto



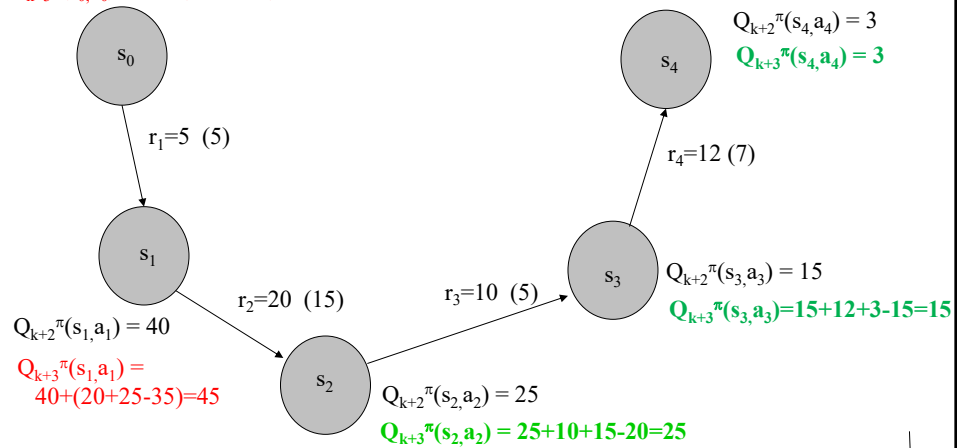
26

Learning $Q^{\pi}(s,a)$ - Trial 3

s_0 = ufficio; s_5 = casa.

$Q_{k+2}^{\pi}(s_0,a_0) = 40$

$Q_{k+3}^{\pi}(s_0,a_0) = 40+(5+40-40)=45$



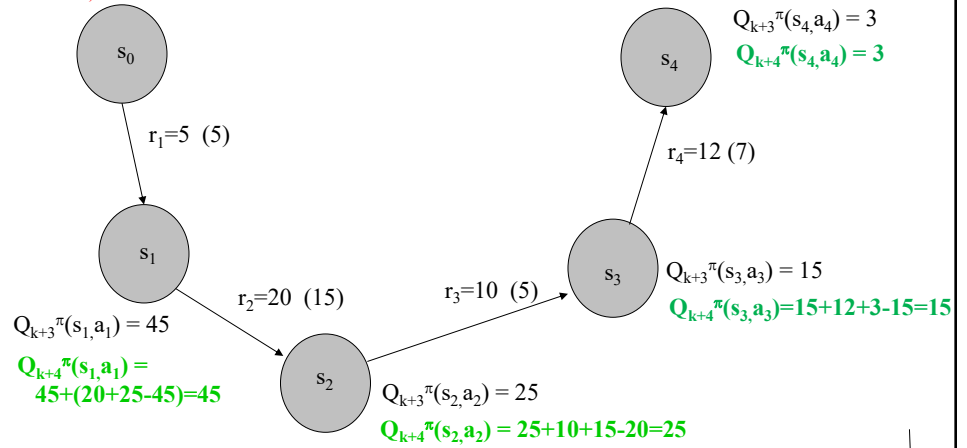
27

Learning $Q^{\pi}(s,a)$ - Trial 4

s_0 = ufficio; s_5 = casa.

$Q_{k+3}^{\pi}(s_0,a_0) = 45$

$Q_{k+4}^{\pi}(s_0,a_0) = 45+(5+35-45)=50$



28

Osservazioni

L'apprendimento procede dalla fine verso l'inizio!

Non posso sapere all'inizio come si concluderà l'episodio, a ogni passo devo cercare di ottenere una stima migliore del reward che otterrò in futuro (per potere poi scegliere)

La stima di $Q^{\pi}(s_{0,}a_0)$ migliora con i trial:

$Q_k^{\pi}(s_{0,}a_0) = 35$
$Q_{k+1}^{\pi}(s_{0,}a_0) = 40$
$Q_{k+2}^{\pi}(s_{0,}a_0) = 45$
$Q_{k+3}^{\pi}(s_{0,}a_0) = 50$

E migliora con il miglioramento della stima del valore dello stato successivo: $Q^{\pi}(s_{1,}a_1)$

Ruolo di α

$(\alpha < 1)$

$Q_{k+1}(s_1,a_1) = Q_k(s_1,a_1) + \alpha\,(r_1 + \gamma Q(s_2,a_2) - Q(s_2,a_2)) = 30+\alpha\,(20+15-30) = 30+\alpha*5$

Stima iniziale del tempo di percorrenza dal parcheggio: 35 m
Tempo per percorrere l'autostrada: 20m
Stima del tempo di percorrenza dall'uscita del parcheggio: 35min (per $\alpha = 1$)

If $\alpha << 1$ aggiorno molto lentamente la value function.

If $\alpha = 1/k(s,a)$ aggiorno la value function in modo da tendere al valore atteso. Devo memorizzare le occorrenze della coppia stato-azione s,a.

If α = cost. Aggiorno la value function, pesando maggiormente i risultati collezionati dalle visite dello stato più recenti.

La convergenza è garantita per α che decresce gradualmente verso zero.

Sommario

Le equazioni di Bellman

Differenze temporali

SARSA

31

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):

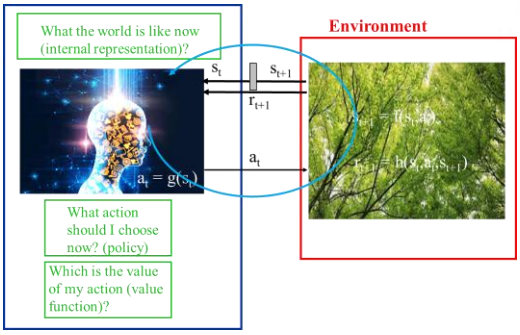
1) Implemento una policy ($\pi(s,a)$)
2) Calcolo la Value function ($Q^{\pi}(s,a)$)
3) Aggiorno la policy.

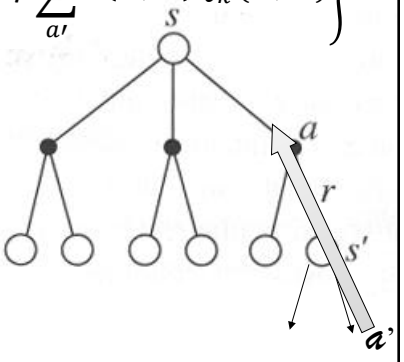
32

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

33

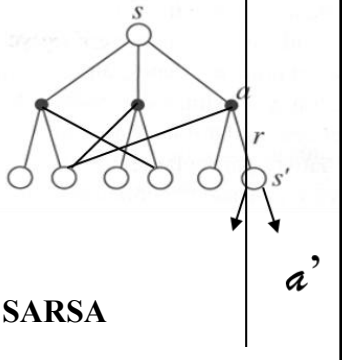
SARSA

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

.

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

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



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

34

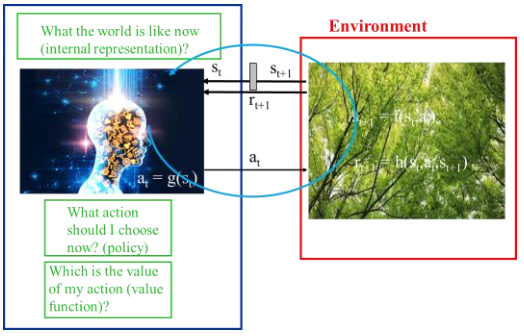
Background su SARSA

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: $P_{s \to s'|a_j}$

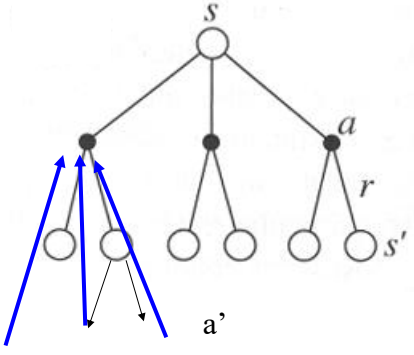
Dopo la realizzazione di un evento, l'incertezza statistica scompare.
1 Reward certo
1 Transizione certa
vengono forniti dall'ambiente



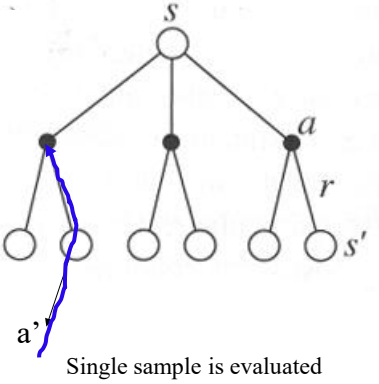
35

Sample backup

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



Full backup



Single sample is evaluated

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

36

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.

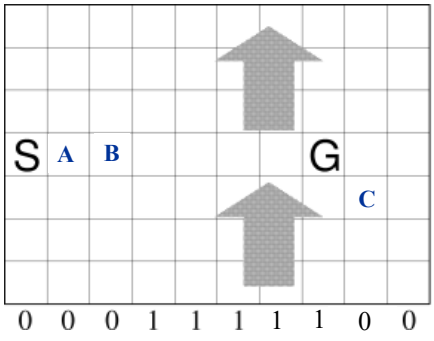
Dovremo poi riapprendere il valore di Q(.)

Come integrare i due passi?

37

Esempio

From Start, S, to Goal, G.



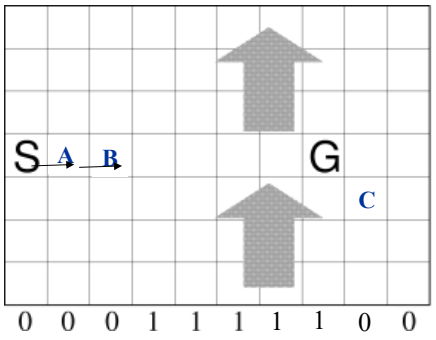
Stati = {caselle della griglia}
Stato iniziale = S
Terminal state = G
Azioni = {su, destra, giù, sinistra}
Reward = -1 tranne che quando s' = TS (reward = 0)

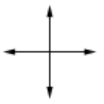
Upwards wind: somma uno spostamento verso l'alto allo spostamento dell'agente nelle prime sei righe del labirinto.

38

Esempio

From Start, S, to Goal, G.





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

Upwards wind

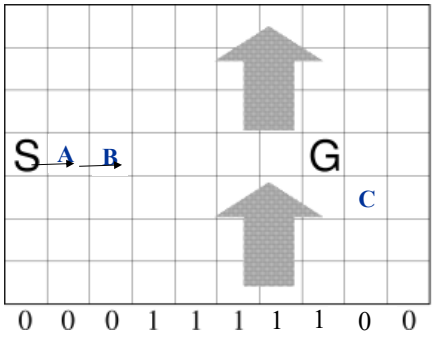
Q(s,a) iniziale = 0.
r = 0 se s' = G; altrimenti r = -1.
π(s,a) data. Azioni possibili: N, S, E, W, NE, NO, SE, SO

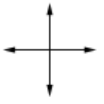
Vogliamo valutare π(s,a). $Q_{k+1}^{\pi}(s,a) = Q_k^{\pi}(s,a) + \alpha[r' + \gamma Q_k^{\pi}(s',a') - Q_k^{\pi}(s,a)]$

39

Esempio - risultato

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





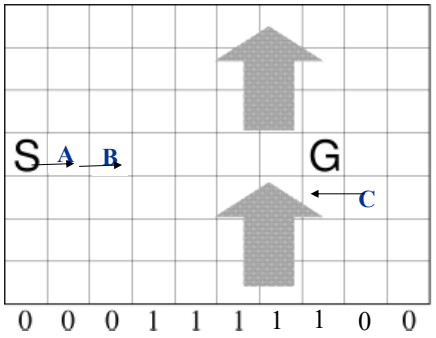
Correzione di Q ad un passo:
$Q_{k+1}^{\pi}(S, east) = Q_k^{\pi}(S, east) + \alpha[r' + \gamma Q_k^{\pi}(A, east) - Q_k^{\pi}(S, east)] = 0 + 0.5\ [-1+1x0-0] = -0.5$

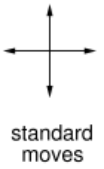
$$Q_{k+1}^{\pi}(s,a) = Q_k^{\pi}(s,a) + \alpha[r' + \gamma Q_k^{\pi}(s',a') - Q_k^{\pi}(s,a)]$$

40

Esempio - risultato

α= 0.5
γ = 1





$Q^{\pi}_{k+1}(A, east) = -0{,}5$

Correzione di Q ad un passo:

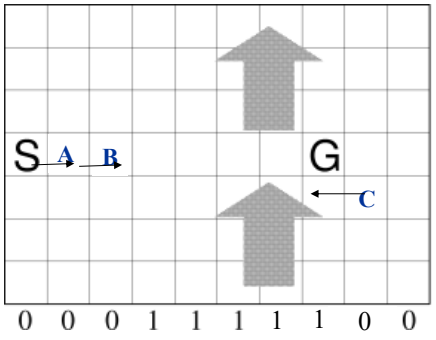
$Q^{\pi}_{k+2}(S, east) = Q^{\pi}_{k+1}(S, east) + \alpha[r' + \gamma Q^{\pi}_{k+1}(A, east) - Q^{\pi}_{k+1}(S, east)] =$

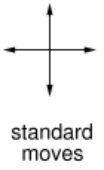
-0.5 + 0.5 [-1+1x(-0.5) – 0.5] = -1

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

Esempio - risultato

α= 0.5
γ = 1





Correzione di Q ad un passo:

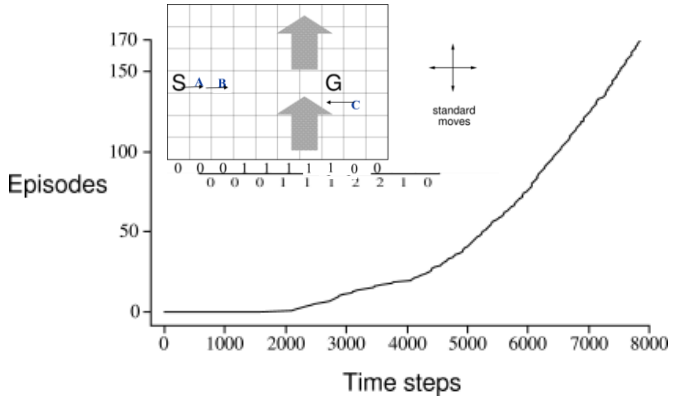
$Q^{\pi}_{k+1}(A, east) = Q^{\pi}_{k}(A, east) + \alpha[r' + \gamma Q^{\pi}_{k}(B, east) - Q^{\pi}_{k}(A, east)] = 0 + 0.5\ [-1+1x0–0] = -0.5$

$Q^{\pi}_{k+1}(C, west) = Q^{\pi}_{k}(C, west) + \alpha[r' + \gamma Q^{\pi}_{k}(G, .) - Q^{\pi}_{k}(C, west)] = 0 + 0.5\ [0+1x0–0] = 0$

(NB c'è il vento verso l'alto di 1)

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

Esempio - risultato



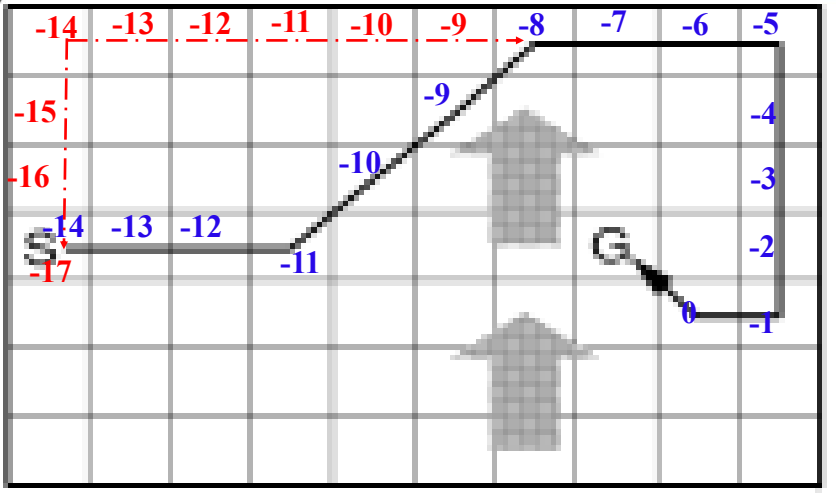
Policy ε-greedy
ε = 0.1
α= 0.5
γ = 1

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.

43

Esempio – valore di $Q^{\pi}(.)$

α= 1 γ = 1

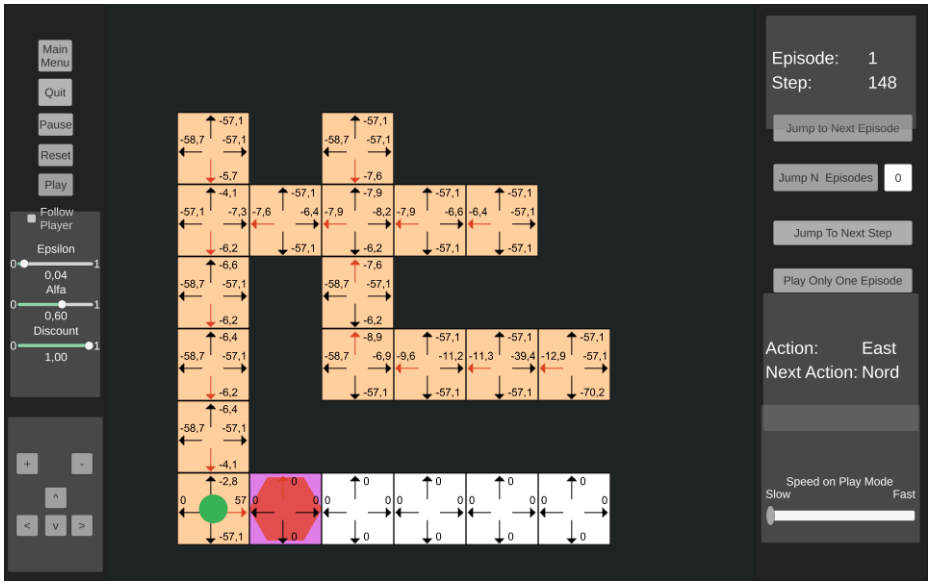


We have estimated $Q^{\pi}(s,a)$. In most states we have better option.

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 $Q^{\pi}(.)$.

44

Labirinto (S. Abate)



Sommario

Equazioni di Bellman

Temporal differences

SARSA