

Sistemi Intelligenti Reinforcement Learning: Temporal Differences

Alberto Borghese

Università degli Studi di Milano
Laboratorio di Sistemi Intelligenti Applicati (AIS-La)
Dipartimento di Informatica
alberto.borghese@unimi.it
Barto and Sutton, Capitoli 3 e 6



A.A. 2024-2025

1/45

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



World RL competition



<https://neurips.cc/Conferences/2024/CompetitionTrack>

Started in NIP2006.

Many competitions, some are based on RL

Other competitions are organized

A.A. 2024-2025

2/45

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



Sommario



Le equazioni di Bellman

Differenze temporali

SARSA



The RL updated picture



Agent

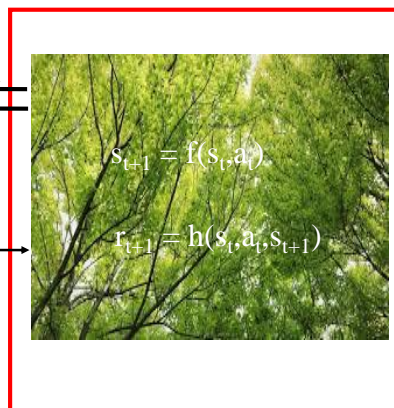
What the world is like now
(internal representation)?



What action
should I choose
now? (policy)

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

Environment



a_t , dipende dalla situazione!



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) Aggiorno la Value function ($Q^\pi(s,a)$)**
- 3) Aggiorno la policy.

Fino a quando la policy non è stabile.

Policy iteration.



Apprendimento



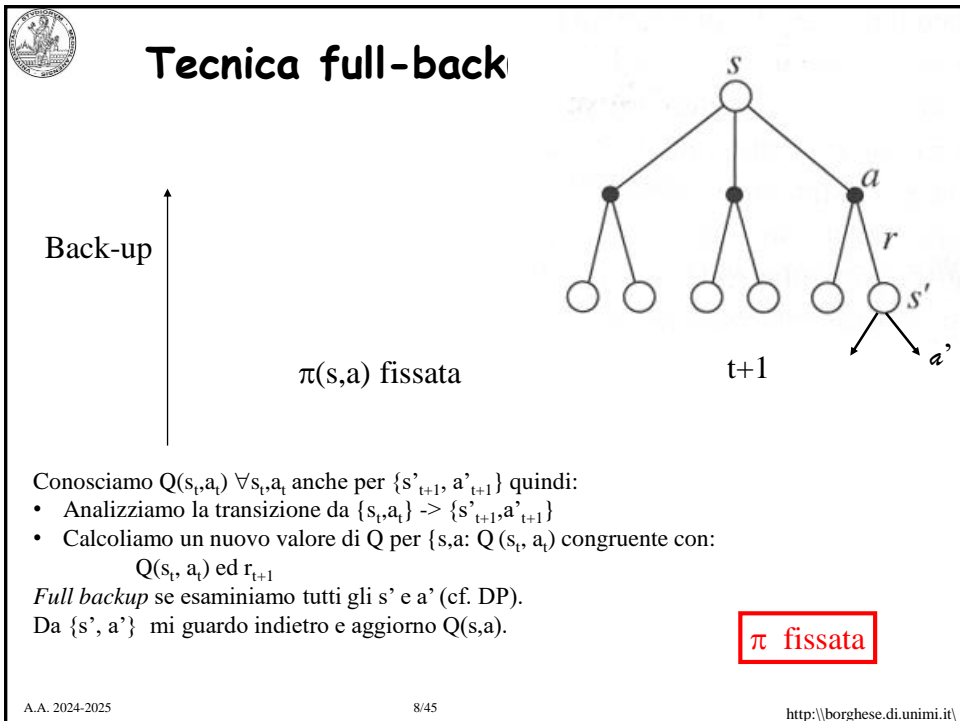
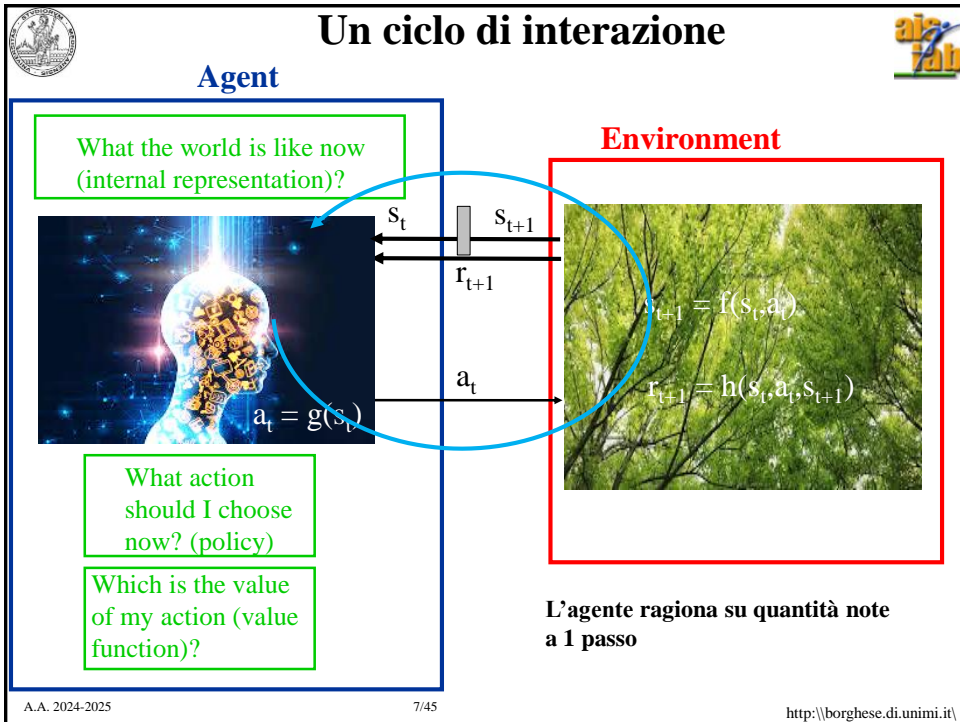
Iterative policy evaluation

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

Converge al limite a $Q^\pi(s,a)$.

Policy improvement

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





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 $Q^\pi(\cdot)$.
2. Stocastico con aggiornamento. **L'agente conosce la statistica della dinamica dell'ambiente e dei reward. Procedo da uno stato iniziale a quello finale. Da ogni stato esplora in parallelo tutti i possibili stati prossimi** e aggiorna i valori delle $Q^\pi(\cdot)$. **Iterative policy evaluation.**
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. Procedo da uno stato iniziale a quello finale. Da ogni stato esplora una sola azione e un SOLO solo stato prossimo.** Aggiorna i valori di $Q^\pi(\cdot)$.



Sommario



Le equazioni di Bellman

Differenze temporali

SARSA

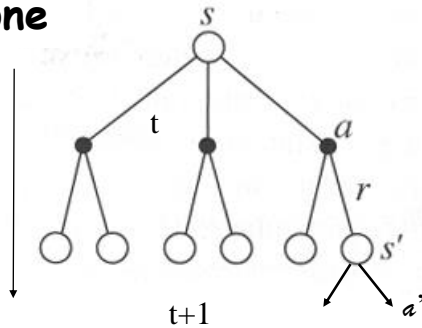


1 passo di interazione

Back-up ↑

π fissata

$\pi(s,a)$ fissata



Analizziamo la transizione da $\{s_t, a_t\} \rightarrow \{s'_{t+1}, a'_{t+1}\}$. 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} : $Q(s'_{t+1}, a'_{t+1}) \forall s'$. *Full backup* se esaminiamo tutti gli s' e a' (cf. DP).

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 e considero il reward a lungo termine SOLO di 1 stato prossimo.



$Q(s, a)$ - Osservazioni



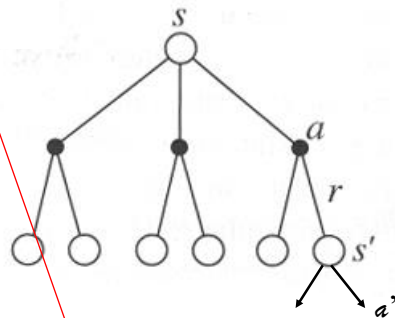
$$Q^\pi(s_t, a_t) = \sum_{s'} P_{s \rightarrow s' | a} \left\{ R_{s, s', a} + \gamma \sum_{a'} \pi(s', a') Q^\pi(s', a') \right\}$$

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)



Background su Temporal Difference (TD) Learning



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

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



Confronto con il rinforzo classico



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

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 = $\alpha = 1/(N+1)$

$a = cost$

Rewards weight $w = 1$

Weight of i -th reward at time k : $w = (1-\alpha)^{k-i}$

Qual è la differenza introdotta dall'approccio che prevede comportamenti (catene di azioni)?

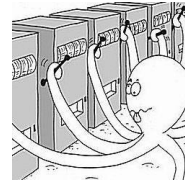


Un possibile aggiornamento di $Q(s, a)$



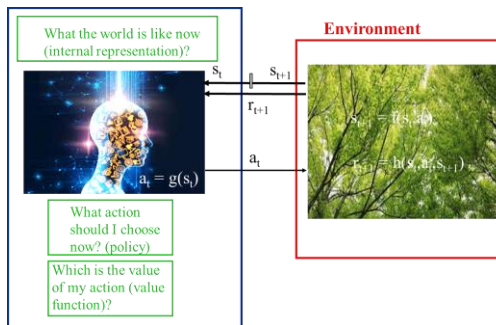
$$Q_{k+1}(a) = Q_k(a) - \frac{Q_k(a)}{N_{k+1}(a)} + \frac{r_{k+1}(a)}{N_{k+1}(a)} = Q_k(a) + \alpha[r_{k+1}(a) - Q_k(a)] =$$

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



Come passo ai comportamenti?

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



Come calcolo ΔQ_k ?



Calcolo di ΔQ_k



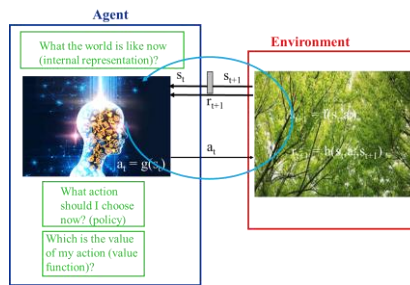
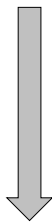
$$Q_{k+1}(a) = Q_k(a) + \alpha[r_{k+1}(a) - Q_k(a)] =$$

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

Al tempo t abbiamo a disposizione:

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

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



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

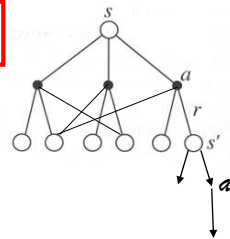
$$Q_{k+1}^\pi(s, a) = Q_k^\pi(s, a) + \alpha[r' + \gamma Q_k^\pi(s', a') - Q_k^\pi(s, a)] =$$

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

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

Reward a 1 passo

Reward a lungo termine da s'



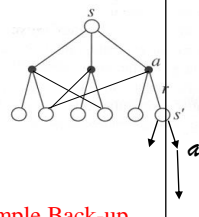


TD(0) update



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

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



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

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

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

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 $k \rightarrow \infty$,

$Q_k^{\pi}(s, a)$ converge al valore vero di $Q^{\pi}(s, a)$

$\pi(s, a)$ fissata

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



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:

$$NewEstimate = OldEstimate + StepSize [Target - OldEstimate]$$

$$NewEstimate = OldEstimate + StepSize * Error.$$

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



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.



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

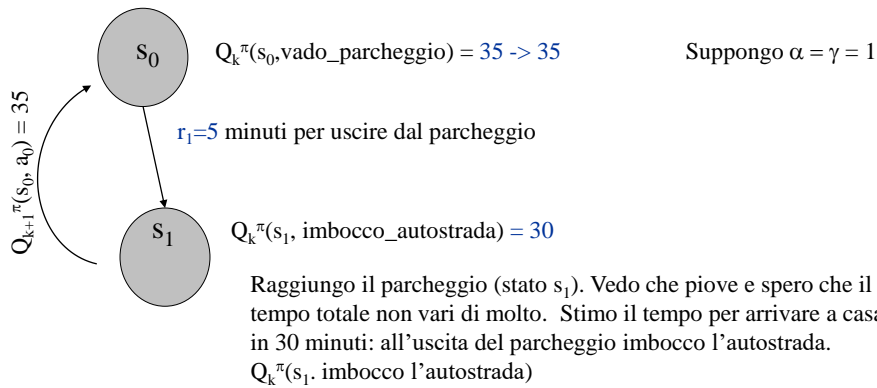
- In totale 35 minuti.



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



s_0 = ufficio; $Q_k^\pi(s_0, \text{vado_parcheggio}) = 35$ minuti; $Q_{k+1}^\pi(s_0, \text{vado_parcheggio}) = 35$ 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).



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

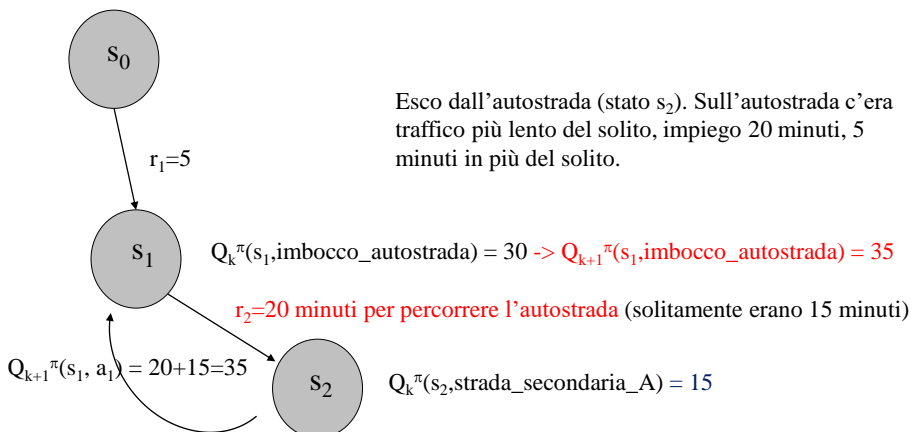
$$Q_{k+1}^\pi(s_0, a) = Q_k^\pi(s_0, a) + \alpha[r' + \gamma Q_k^\pi(s_1, a') - Q_k^\pi(s_0, a)] = 35 + [5 + 30 - 35] = 35$$



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)



Aggiorno il tempo totale dallo stato s_1 :

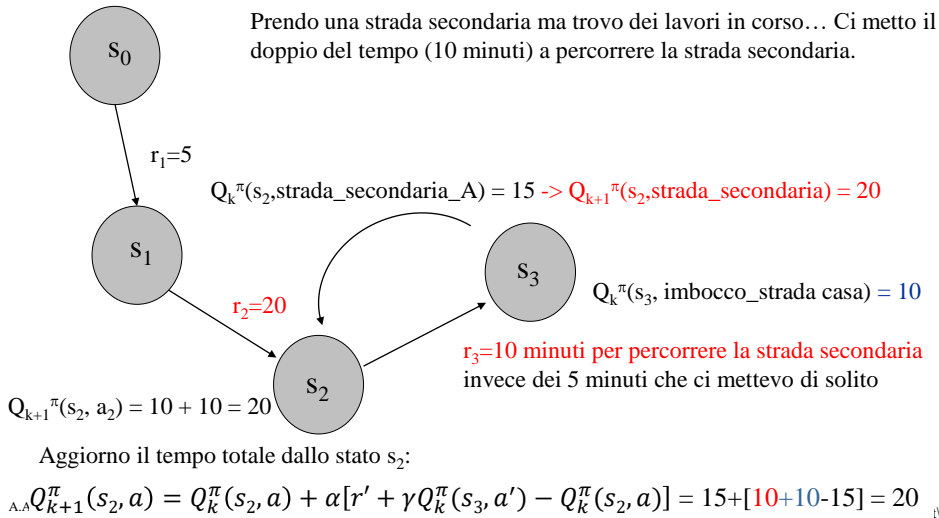
$$Q_{k+1}^\pi(s_1, a) = Q_k^\pi(s_1, a) + \alpha[r' + \gamma Q_k^\pi(s_2, a') - Q_k^\pi(s_1, a)] = 30 + [20 + 15 - 30] = 35$$



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



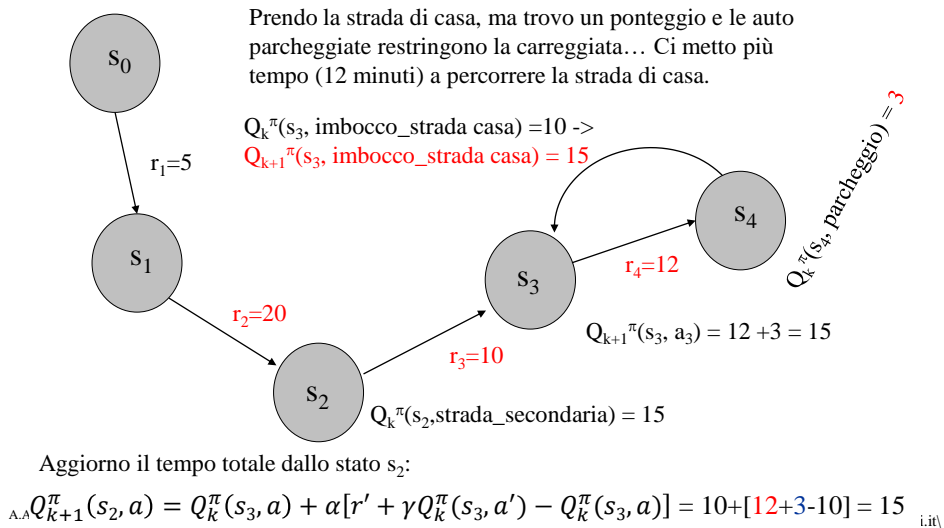
$s_2 = \text{esco_autostrada}$; $Q_k^\pi(s_2, \text{esco_autostrada}) = 15 \text{ min}$; $Q_{k+1}^\pi(s_2, \text{esco_autostrada}) = 20 \text{ min}$;



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



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





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)
- 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 $Q(\cdot)$ è per lo stato prima di quello finale ed è di 15 minuti. La stima di $Q(\cdot)$ 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?



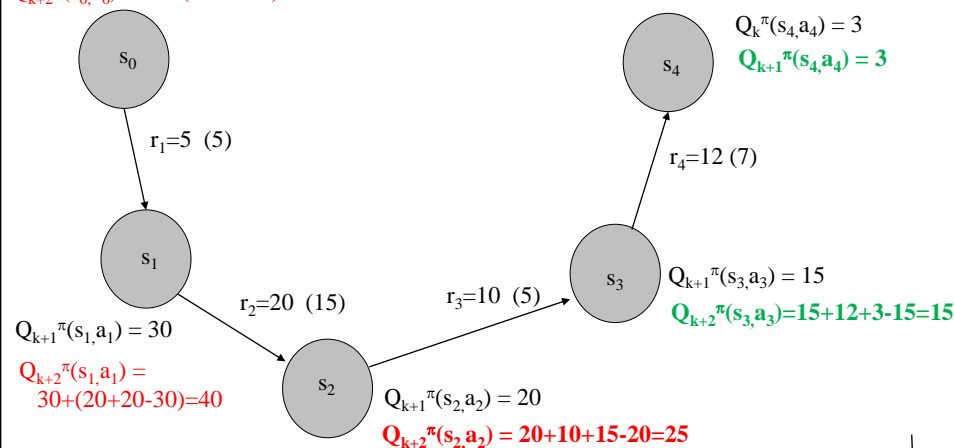
Learning $Q^\pi(s,a)$ - Trial 2





s_0 = ufficio; s_5 = casa.

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

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



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

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

$Q_{k+2}^\pi(s_0, a_0) = 40$
 $Q_{k+3}^\pi(s_0, a_0) = 40 + (5 + 40 - 40) = 45$

$Q_{k+2}^\pi(s_1, a_1) = 40$
 $Q_{k+3}^\pi(s_1, a_1) = 40 + (20 + 25 - 35) = 45$



$Q_{k+2}^\pi(s_2, a_2) = 25$
 $Q_{k+3}^\pi(s_2, a_2) = 25 + 10 + 15 - 20 = 25$

$Q_{k+2}^\pi(s_3, a_3) = 15$
 $Q_{k+3}^\pi(s_3, a_3) = 15 + 12 + 3 - 15 = 15$

$Q_{k+2}^\pi(s_4, a_4) = 3$
 $Q_{k+3}^\pi(s_4, a_4) = 3$

Diagram showing states s_0, s_1, s_2, s_3, s_4 and transitions with rewards $r_1=5, r_2=20, r_3=10, r_4=12$.

A.A. 2024-2025 27/45 <http://borghese.di.unimi.it/>

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

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

$Q_{k+3}^\pi(s_0, a_0) = 45$
 $Q_{k+4}^\pi(s_0, a_0) = 45 + (5 + 35 - 45) = 50$

$Q_{k+3}^\pi(s_1, a_1) = 45$
 $Q_{k+4}^\pi(s_1, a_1) = 45 + (20 + 25 - 45) = 45$

$Q_{k+3}^\pi(s_2, a_2) = 25$
 $Q_{k+4}^\pi(s_2, a_2) = 25 + 10 + 15 - 20 = 25$

$Q_{k+3}^\pi(s_3, a_3) = 15$
 $Q_{k+4}^\pi(s_3, a_3) = 15 + 12 + 3 - 15 = 15$

$Q_{k+3}^\pi(s_4, a_4) = 3$
 $Q_{k+4}^\pi(s_4, a_4) = 3$

Diagram showing states s_0, s_1, s_2, s_3, s_4 and transitions with rewards $r_1=5, r_2=20, r_3=10, r_4=12$.

A.A. 2024-2025 28/45 <http://borghese.di.unimi.it/>



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 $\alpha \ll 1$ aggrano molto lentamente la value function.

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

If $\alpha = \text{cost}$. Aggrano 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



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):

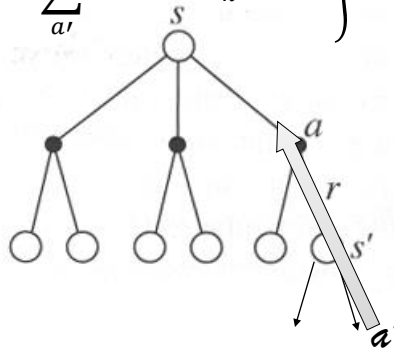
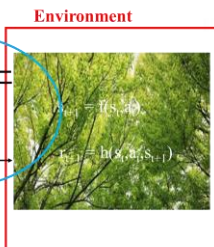
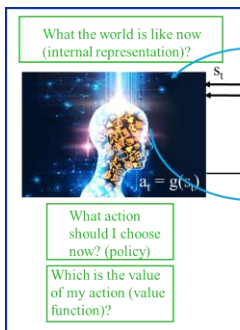
- 1) Implemento una policy ($\pi(s,a)$)
- 2) **Calcolo la Value function ($Q^\pi(s,a)$)**
- 3) Aggiorno la policy.



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(.)



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



SARSA

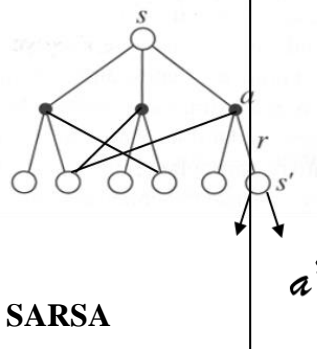
Non richiede conoscenze a priori dell'ambiente.

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

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

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



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



Background su SARSA



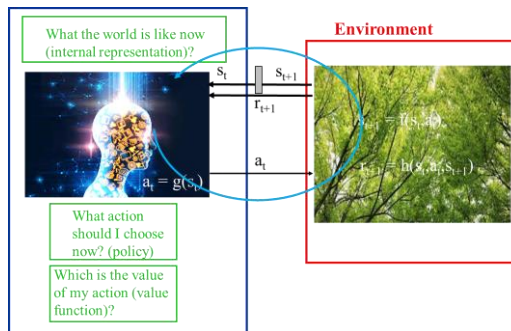
Al tempo t abbiamo a disposizione:

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

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

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

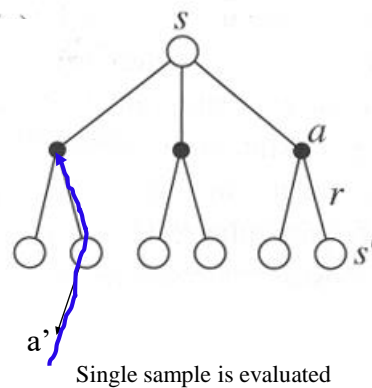
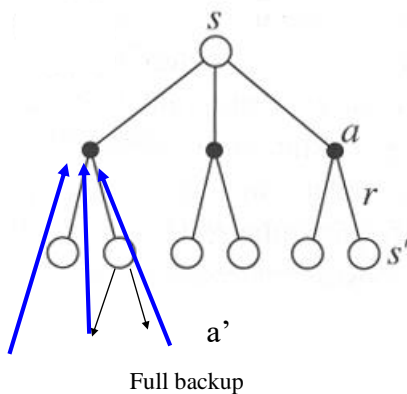
- 1 Reward certo
- 1 Transizione certa vengono forniti dall'ambiente



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'



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(.).

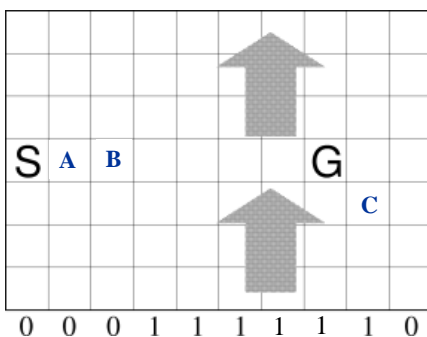
Come integrare i due passi?



Esempio



From Start, S, to Goal, G.

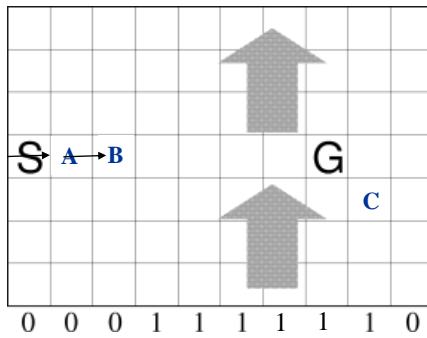


Stati = {caselle della griglia}
 Stato iniziale = S
 Terminal state = G
 Azioni = {su, destra, giù, sinistra}
 Reward = -1 tranne che quando s' = TS (reward = 0)

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



Esempio



from Start, S, to Goal, G.

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

Upwards wind

$Q(s,a)$ iniziale = 0.

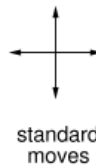
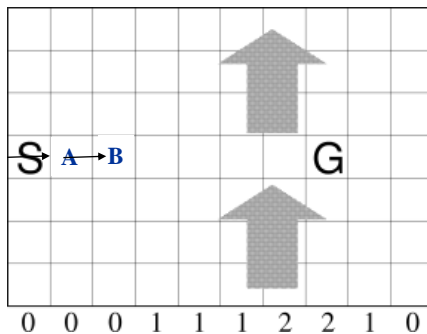
$r = 0$ se $s' = G$; altrimenti $r = -1$.

$\pi(s,a)$ data. Azioni possibili: N, S, E, W, NE, NO, SE, SO

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



Esempio - risultato



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

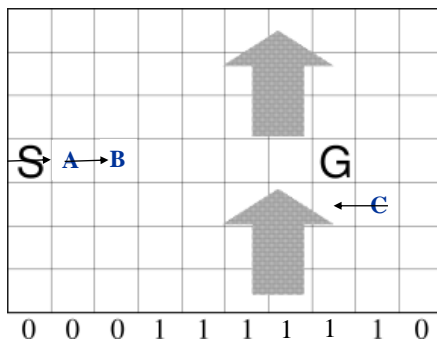
Correzione di Q ad un passo:

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

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



Esempio - risultato



$$\alpha = 0.5$$

$$\gamma = 1$$



$$Q_{k+1}^{\pi}(A, east) = -0.5$$

Correzione di Q ad un passo:

$$Q_{k+2}^{\pi}(S, east) = Q_{k+1}^{\pi}(S, east) + \alpha[r' + \gamma Q_{k+1}^{\pi}(A, east) - Q_{k+1}^{\pi}(S, east)] = -0.5 + 0.5[-1 + 1 \times (-0.5) - 0.5] = -1$$

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

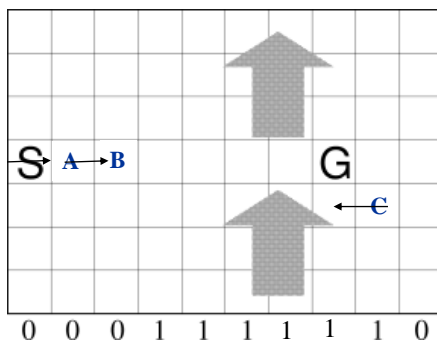
A.A. 2024-2025

41/45

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

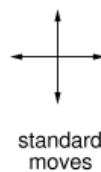


Esempio - risultato



$$\alpha = 0.5$$

$$\gamma = 1$$



Correzione di Q ad un passo:

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

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

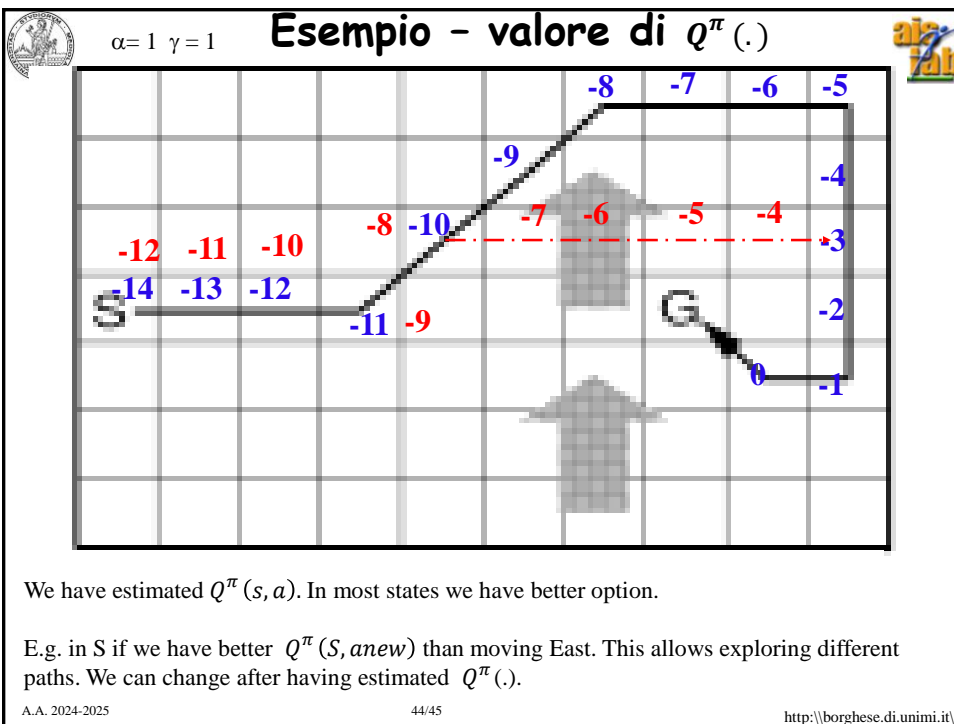
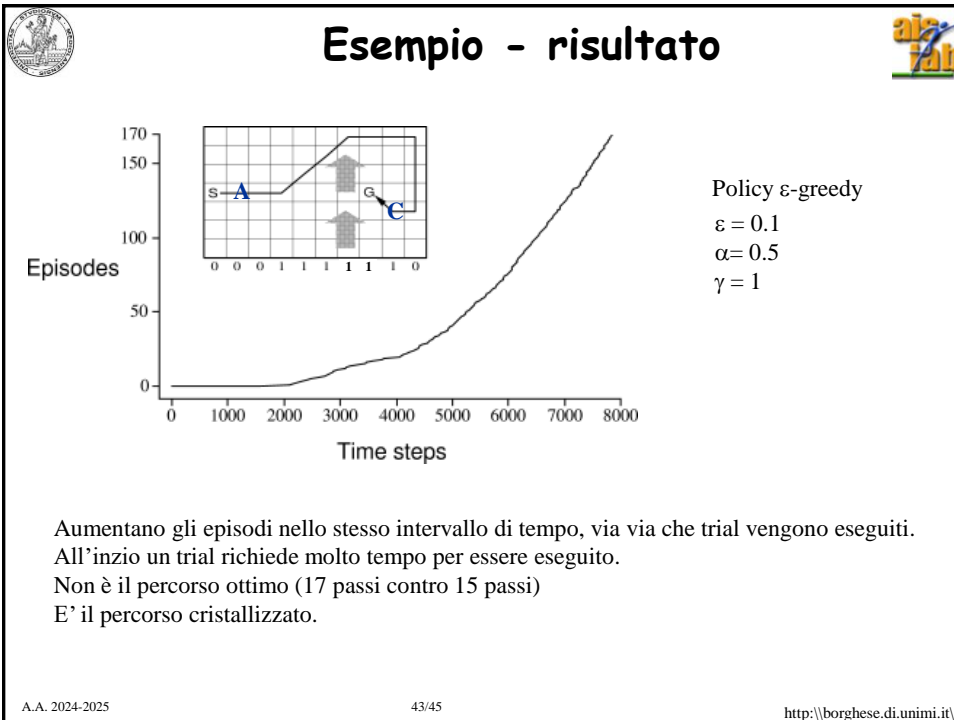
(NB c'è il vento verso l'alto di 1)

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

A.A. 2024-2025

42/45

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





Sommario



Equazioni di Bellman

Temporal differences

SARSA