

Sistemi Intelligenti Reinforcement Learning: Temporal Difference

Alberto Borghese

Università degli Studi di Milano
Laboratorio di Sistemi Intelligenti Applicati (AIS-Lab)
Dipartimento di Scienze dell'Informazione
borghese@di.unimi.it



A.A. 2013-2014

1/28



Sommario

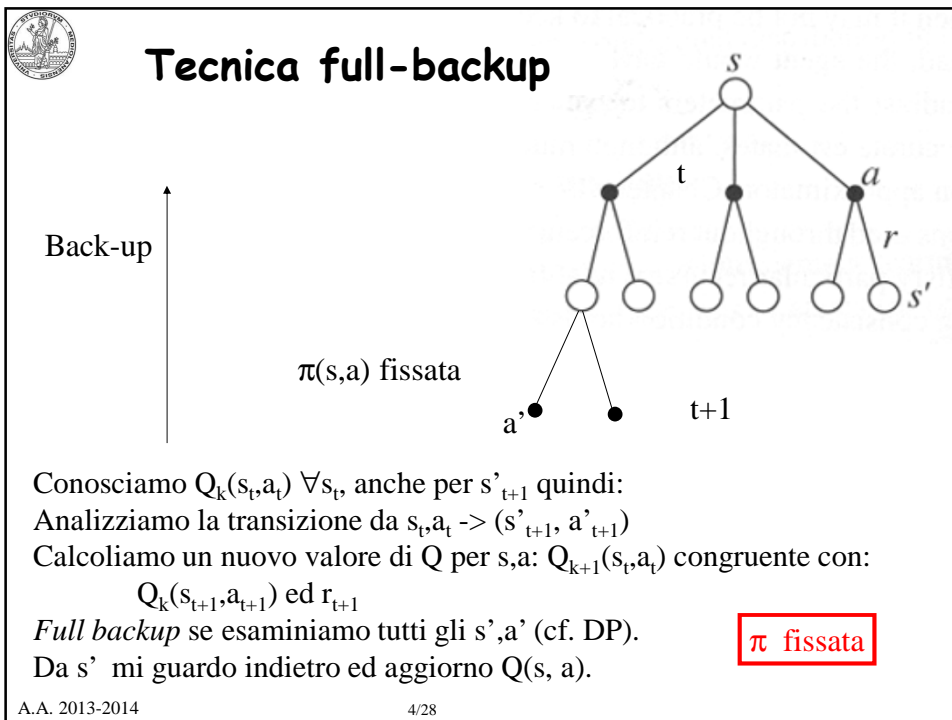
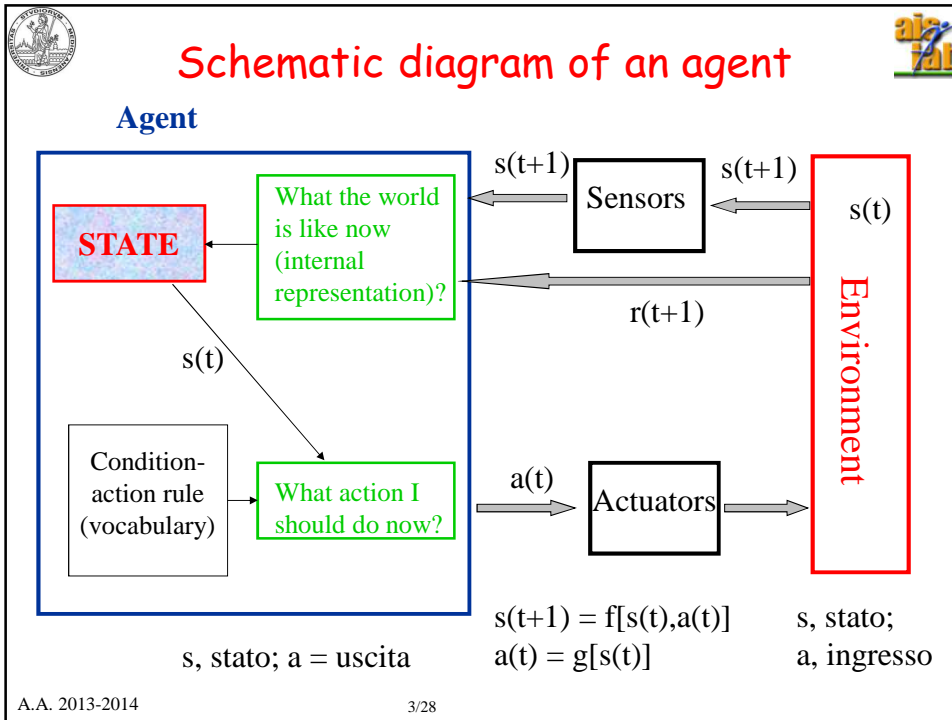


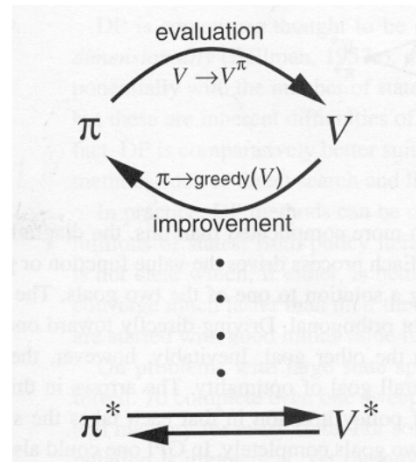
Temporal differences

SARSA

A.A. 2013-2014

2/28





Generalized Policy iteration

{ Policy iteration
Value iteration

Schema di Apprendimento

Competizione e cooperazione -> V corretta e policy ottimale.



World RL competition

Started in NIP2006.

It became very popular and started a workshop on its own. Visit:

<http://rl-competition.org>



How About Learning the Value Function?



Facciamo imparare all'agente la value function, per una certa politica: V^π :

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

È una funzione dello stato.

Una volta imparata la value function, Q^* , l'agente seleziona la policy ottima passo per passo, "one step lookahead":

$$\pi^*(s) = \arg \max_{a'} \sum_{s'} P_{s \rightarrow s' | a}^a [R_{s \rightarrow s' | a}^a + \gamma V^\pi(s')] = \arg \max_{a'} \sum_{s'} P_{s \rightarrow s' | a}^a [R_{s \rightarrow s' | a}^a + \gamma \pi(s', a') Q^\pi(s', a')]$$

Full backup, for all states



Value iteration



Facciamo imparare all'agente la value function, per una certa politica: Q^π , analizzando quello che succede in uno step temporale:

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

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

L'apprendimento della policy si può inglobare nella value iteration:

$$Q_{k+1}(s, a) = \max_{a'} \sum_{s'} P_{s \rightarrow s' | a} [R_{s \rightarrow s' | a} + \gamma Q_k(s', a')] \quad \forall s, a$$



Problema legato alla conoscenza della risposta dell'ambiente



$$Q_{k+1}(s, a) \leftarrow \max_a \sum_{s'} P_{s \rightarrow s'|a} \left[R_{s \rightarrow s'|a} + \gamma \sum_{a'} \pi(s', a') Q_k(s', a') \right]$$

Full backup, single state, s, all future states s'

Fino a questo punto, è noto un modello dell'ambiente:

- R(.)
- P(.)

Environment modeling -> Value function computation -> Policy optimization.



Osservazioni



Iterazione tra:

- Calcolo della Value function

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

- Miglioramento della policy

$$= \arg \max_{a'} \sum_{s'} P_{s \rightarrow s'|a}^a \left[R_{s \rightarrow s'|a}^a + \gamma Q_k^\pi(s', a') \right]$$

Non sono noti



Background su Temporal Difference (TD) Learning



Al tempo t abbiamo a disposizione:

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

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

Reward certo
Transizione certa
vengono misurati dall'ambiente

Come si possono utilizzare per apprendere?



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:

$$\text{NewEstimate} = \text{OldEstimate} + \text{StepSize} [\text{Target} - \text{OldEstimate}]$$

$$\text{NewEstimate} = \text{OldEstimate} + \text{StepSize} * \text{Error}.$$

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

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

Qual è la differenza introdotta dall'approccio DP?



Un possibile aggiornamento

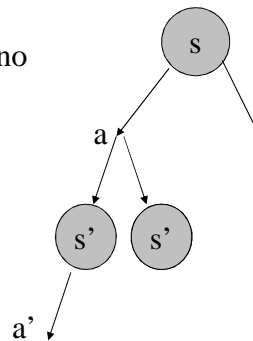


In iterative policy evaluation ottengo questo aggiornamento:

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

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

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



Qual'è il problema?

A.A. 2013-2014

13/28



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



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

Quanto vale α ?

$$Q_k(s, a) = Q_k(s, a) + \Delta Q_k(s, a)$$

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

A.A. 2013-2014

14/28



TD(0) update

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

$$Q_{k+1}(s_t, a_t) = Q_k(s_t, a_t) + \alpha [r_{t+1} + \gamma Q_k(s_{t+1}, a_{t+1}) - Q_k(s_t, a_t)]$$

Da confrontare con la iterative policy evaluation:

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

Sample backup

E con il valore di uno stato sotto la policy $\pi(s, a)$:

$$Q^\pi(s, a) = E_\pi \{ R_t | s_t = s, a_t = a \} = E_\pi \{ r_{t+1} + \gamma Q^\pi(s', a') | s_t = s, a_t = a \}$$

Quanto vale α ?



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:

$$\begin{aligned} \text{NewEstimate} &= \text{OldEstimate} + \text{StepSize} [\text{Target} - \text{OldEstimate}] \\ \text{NewEstimate} &= \text{OldEstimate} + \text{StepSize} * \text{Error} \end{aligned}$$

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



Setting α value

$\alpha(s_t, a_t, s_{t+1}) = 1/k(s_t, a_t, s_{t+1})$, where k 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 da (s,a) , con peso: $(1-\alpha)^k$: *exponential recency-weighted average*.



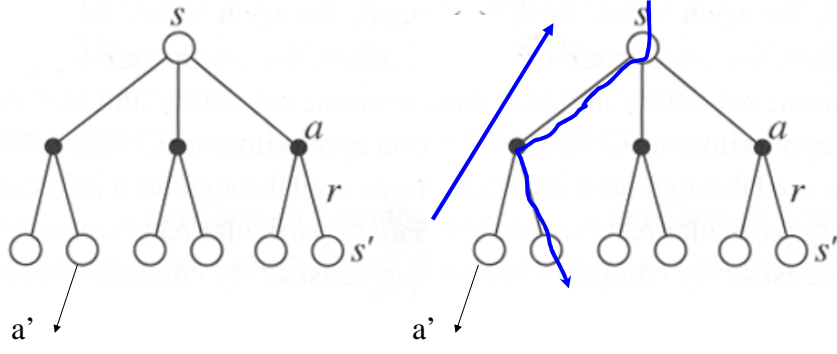
Sommario

Temporal differences

SARSA




Sample backup



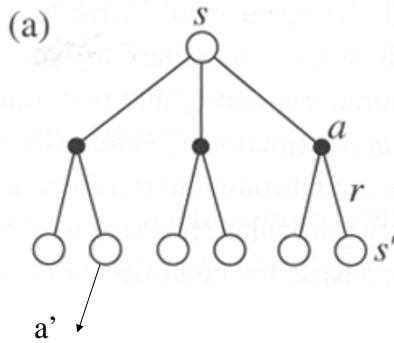
Full backup

Single sample is evaluated

A.A. 2013-2014 19/28



Sample backup





(a)

SARSA algorithm

- State –
- Action –
- Reward –
- State (next) –
- Action (next)

A.A. 2013-2014 20/28






Algoritmo per TD(0) - Progetto per esame (da completare con scelta della policy)

Inizializziamo $Q(s,a) = 0$.
 Inizializziamo la policy: $\pi(s,a)$. da valutare

```
Repeat
{
  s = s0; a =  $\pi(s)$ ;
  Repeat // For each state until terminal state, analyze an episode
  {
    s_next = NextState(s, a);
    reward = Reward(s, s_next, a);
    a_next =  $\pi(s\_next)$ 
     $Q(s,a) = Q(s,a) + \alpha [\text{reward} + \gamma Q(s\_next,a\_next) - Q(s,a)]$ ;
    s = s_next; a = a_next;
  } until TerminalState
} Until convergence of  $Q(s,a)$  for policy  $\pi(s,a)$ 
```

A.A. 2013-2014 21/28

Esempio: valutazione della policy TD

Stato	Tempo trascorso attuale	Tempo percorrenza attuale del segmento	Tempo percorrenza a stimato fino ad ora del segmento	Tempo totale previsto in precedenza $- Q_k(s',a')$	Tempo totale previsto aggiornato $- Q_{k+1}(s,a)$	Increase or decrease $-Q(s,a)$
Esco dall'ufficio	0	0	0	43	$43 + \alpha(5 + 38 - 43)$	=
Salgo in auto	5	5	5	38	$38 + \alpha(10 + 23 - 38)$	<
Esco dall'autostrada	15	10	15	23	$23 + \alpha(7 + 18 - 23)$	>
Strada secondaria con camion davanti	22	7	5	18	$18 + \alpha(10 + 13 - 18)$	>
Strada di casa	32	10	5	13	$13 + \alpha(10 + 3 - 13)$	=
Entro in casa	42	10	10	3	$3 + \alpha(5 + 3 - 3)$	>
	47	5	3	0	0	

$Q(s,a)$ è l'expected "Time-to-Go" - $\gamma = 1$

A.A. 2013-2014 22/28



Ruolo di α



$$Q(1, a_1) = (1, a_1) + \alpha (r_1 + \gamma(2, a_2) - Q(1)) = 38 + \alpha (10 + 23 - 38) = 38 - \alpha * 5$$

Stima iniziale del tempo di percorrenza dal parcheggio: 38m

Tempo di percorrenza fino ad uscita autostrada: 10m

Stima del tempo di percorrenza dall'uscita autostrada: 23m

$\alpha < 1$.

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

If $\alpha = 1/k$ aggiorno la value function in modo da tendere al valore atteso. Devo memorizzare le occorrenze dello stato s .

If $\alpha = \text{cost}$. Aggiorno la value function, pesando maggiormente i risultati collezionati dalle visite dello stato più recenti.

A.A. 2013-2014

23/28



Proprietà del metodo TD



Non richiede conoscenze a priori dell'ambiente.

L'agente stima dalle sue stesse stime precedenti (bootstrap).

Si dimostra che il metodo converge asintoticamente.

Batch vs trial learning.

Converge!!

$$Q^\pi(s_t, a_t) = Q^\pi(s_t, a_t) + \alpha [r_{t+1} + \gamma Q^\pi(s_{t+1}, a_{t+1}) - Q^\pi(s_t, a_t)]$$

Single backup, single state, s_t , single future state s_{t+1}

Rimpiazza iterative Policy evaluation.

Rimane il passo di Policy iteration (improvement).

A.A. 2013-2014

24/28



SARSA Algorithm (progetto)

```

Q(s,a) = rand(); // ∀s, ∀a, eventualmente Q(s,a) = 0
Repeat
{
  s = s0;
  Repeat // for each step of the single episode
  {
    a = Policy(s); // ε-greedy??
    s_next = NextState(s,a);
    reward = Reward(s,s_next,a);
    a_next = Policy(s_next); // ε-greedy?
    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

```

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

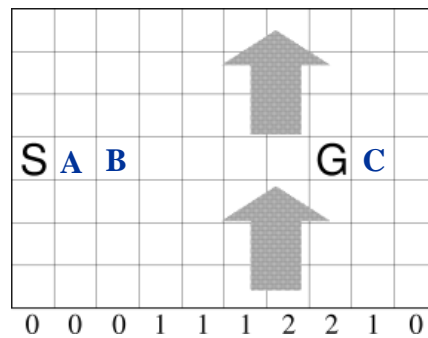
Come integrare i due passi?

A.A. 2013-2014

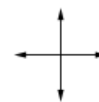
25/28



Esempio



From Start to Goal.



standard moves

Upwards wind

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

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

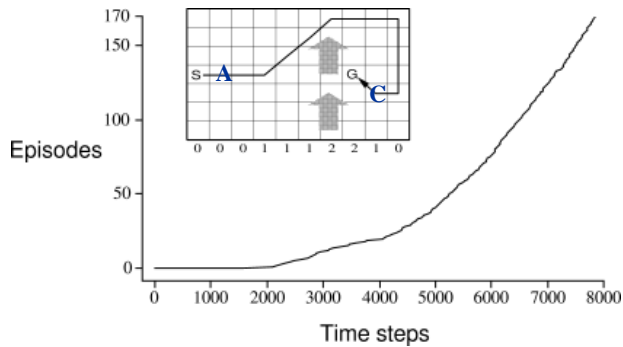
$\pi(s,a)$ data.

A.A. 2013-2014

26/28



Esempio - risultato



Policy π , greedy
or ϵ -greedy

$$\epsilon = 0.1$$

$$\alpha = 0.5$$

$$\gamma = 1$$

Per trial or
per epoch

Al termine,
policy
improvement.

Correzione di Q ad un passo:

$$Q(S, \text{east}) = 0 + 0.5 [-1 + 0 - 0] = -0.5$$

$$Q(A, \text{east}) = 0 + 0.5 [-1 + 0 - 0] = -0.5$$

$$Q(C, \text{west}) = 0 + 0.5 [0 + 0 - 0] = 0; \quad (\text{NB c'è il vento verso l'alto di 1})$$

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha [r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)]$$

A.A. 2013-2014



Sommario



Temporal differences

SARSA

A.A. 2013-2014

28/28