

Sistemi Intelligenti Reinforcement Learning: Eligibility Trace

Alberto Borghese

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



A.A. 2009-2010

1/25

<http://homes.dsi.unimi.it/~borghese/>



Sommario



- The eligibility trace
- $Q(\lambda)$, SARSA(λ)

A.A. 2009-2010

2/25

<http://homes.dsi.unimi.it/~borghese/>



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 assignment.
structural credit assignment.

L'apprendimento non è più da esempi, ma dall'osservazione del proprio comportamento nell'ambiente.



Come apprendere Q: SARSA



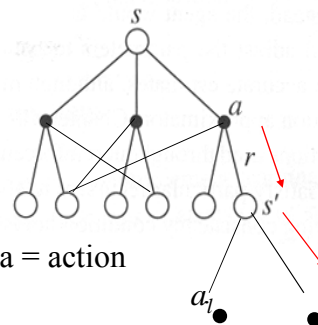
$$Q(s_t, a_t) = Q^\pi(s_t, a_t) + \alpha [r_{t+1} + \gamma Q^\pi(s_{t+1}, a_{t+1}) - Q^\pi(s_t, a_t)]$$

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 (**policy improvement**)

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

On-policy learning.





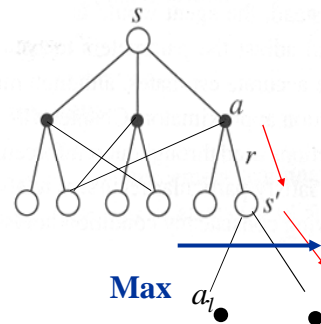
Off-policy Temporal Difference: Q-learning



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

Non imparo semplicemente la funzione valore Q, ma la funzione valore Q ottima.

In s , scelgo un ramo del grafo, e poi **decido** ad un passo come continuare.



A.A. 2009-2010

5/25

<http://homes.dsi.unimi.it/~borghese/>



Formulazione di TD(0)



Correggo la stima corrente valutando l'errore ad un passo.

$$V_{k+1}(s_t) = V_k(s_t) + \alpha [r_{t+1} + \gamma V_k(s_{t+1}) - V_k(s_t)]$$

$$\Delta V(s_t) = + \alpha \delta_k \quad \delta_k = [r_{t+1} + \gamma V_k(s_{t+1}) - V_k(s_t)]$$

Estensione dell'orizzonte temporale dell'apprendimento

A.A. 2009-2010

6/25

<http://homes.dsi.unimi.it/~borghese/>



Cosa rappresenta la Eligibility trace



Buffer di memoria: contiene traccia di eventi passati (stati visitati, azioni...); la traccia evapora nel tempo.

Quando viene calcolato un errore usando metodi basati su TD, la eligibility trace suggerisce quali variabili aggiornare (credit assignment).

Amplia l'orizzonte temporale sul quale fare l'aggiornamento a più di 1 passo.



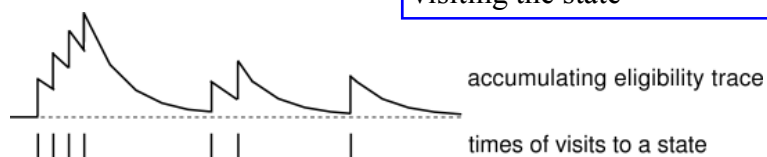
Eligibility trace



$$e_t(s) = \begin{cases} \gamma \lambda e_{t-1}(s) + 1 & \text{If } s = s_t \\ \gamma \lambda e_{t-1} & \text{Otherwise} \end{cases}$$

decay

Increases: depends only on visiting the state



$$e(s) = 0 \text{ at start, } e(s) \geq 0.$$



Formulazione di TD(λ) per $V(s)$



Correggo la stima corrente valutando l'errore ad un passo e distribuendolo su **tutti gli stati**.

$$V_{k+1}(s_t) = V_k(s_t) + \alpha [r_{t+1} + \gamma V_k(s_{t+1}) - V_k(s_t)]$$

$$\Delta V(s_t) = + \alpha \delta_k e(s_t) \quad \delta_k = [r_{t+1} + \gamma V_k(s_{t+1}) - V_k(s_t)]$$

$\forall s_t$

Estensione dell'orizzonte temporale dell'apprendimento



Osservazioni

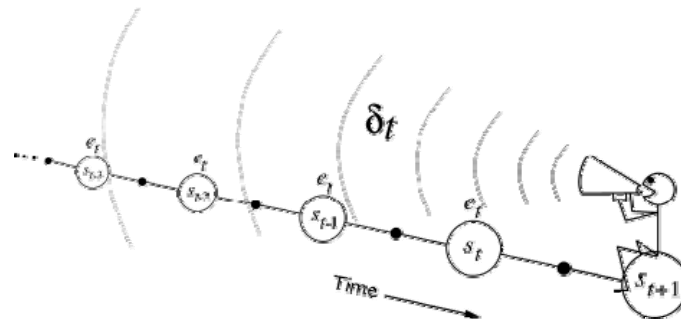


Figure 7.8: The backward or mechanistic view. Each update depends on the current TD error combined with traces of past events.

Earlier states are given less credit for the TD(0) error



Algoritmo per $V(\lambda)$



```

Initialize  $V(s)$  arbitrarily and  $e(s) = 0$ , for all  $s \in \mathcal{S}$ 
Repeat (for each episode):
  Initialize  $s$ 
  Repeat (for each step of episode):
     $a \leftarrow$  action given by  $\pi$  for  $s$ 
    Take action  $a$ , observe reward,  $r$ , and next state,  $s'$ 
     $\delta \leftarrow r + \gamma V(s') - V(s)$ 
     $e(s) \leftarrow e(s) + 1$ 
    For all  $s$ :
       $V(s) \leftarrow V(s) + \alpha \delta e(s)$ 
       $e(s) \leftarrow \gamma \lambda e(s)$ 
     $s \leftarrow s'$ 
  until  $s$  is terminal

```

A.A. 2009-2010

11/25

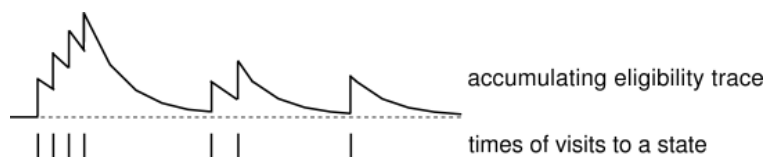
<http://homes.dsi.unimi.it/~borghese/>



Eligibility trace per la funzione Q



$$e_t(s, a) = \begin{cases} \gamma \lambda e_{t-1}(s, a) + 1 & \text{if } s = s_t \text{ and } a = a_t; \\ \gamma \lambda e_{t-1}(s, a) & \text{otherwise.} \end{cases} \quad \text{for all } s, a$$



A.A. 2009-2010

12/25

<http://homes.dsi.unimi.it/~borghese/>



Come utilizzare la eligibility trace



TD(0) Learning:

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

Errore: δ_t

$$Q_{k+1}(s_t, a_t) = Q_k(s_t, a_t) + \alpha \delta_t \quad \text{Per 1 coppia (s,a)}$$

$$Q_{k+1}(s, a) = Q_k(s, a) + \alpha \delta_t e_t(s, a) \quad \text{Per tutte le coppie (s,a)}$$

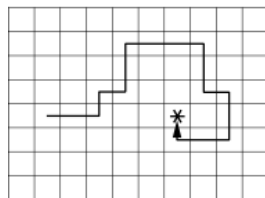
Eleggibilità: $e_t(s, a)$



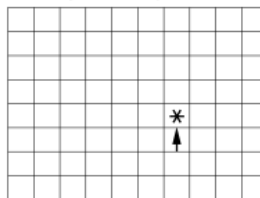
Esempio



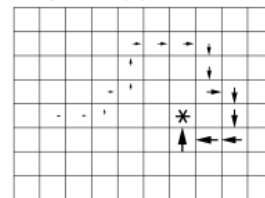
Path taken



Action values increased by one-step Sarsa



Action values increased by Sarsa(λ) with $\lambda=0.9$



Con il semplice costo di una variabile per ogni coppia stato-azione, ho un aggiornamento graduale della funzione valore di più stati.

$Q^\pi(s, a)$ inizializzati ad un valore leggermente negativo.

$r = 0$ per ogni stato prossimo, tranne lo stato finale, per il quale $r = +1$.



SARSA(λ)



Initialize $Q(s, a)$ arbitrarily and $e(s, a) = 0$, for all s, a
Repeat (for each episode):
 Initialize s, a
 Repeat (for each step of episode):
 Take action a , observe r, s'
 Choose a' from s' using policy derived from Q (e.g., ϵ -greedy)
 $\delta \leftarrow r + \gamma Q(s', a') - Q(s, a)$
 $e(s, a) \leftarrow e(s, a) + 1$
 For all s, a :
 $Q(s, a) \leftarrow Q(s, a) + \alpha \delta e(s, a)$
 $e(s, a) \leftarrow \gamma \lambda e(s, a)$
 $s \leftarrow s'; a \leftarrow a'$
 until s is terminal



Sommario



- The eligibility trace
- $Q(\lambda)$



Watkin's version of $Q(\lambda)$



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

Quanto posso propagare all'indietro l'"errore"?

NB L'azione che scelgo può non essere la migliore, la funzione Q viene modificata run-time.



Watkin's $Q(\lambda)$



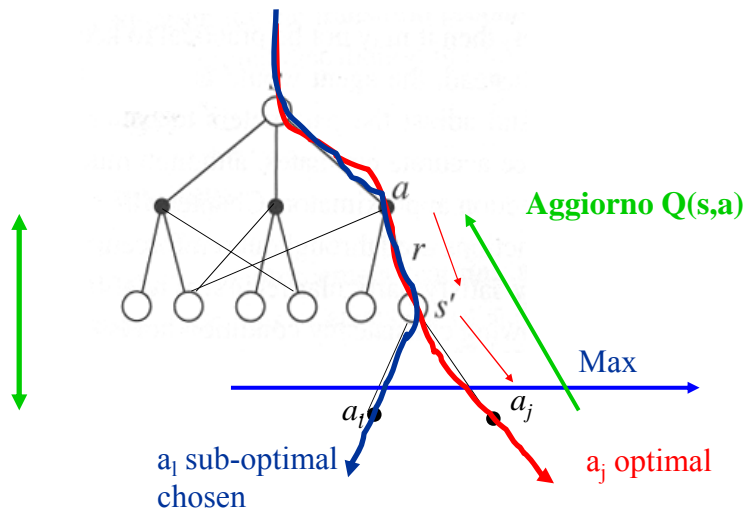
$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left[r_{t+1} + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t) \right]$$

Suppongo di scegliere $a^* = a_{t+1}$ azione esplorativa secondo la policy π .

Posso sempre calcolare $Q(s_t, a_t)$, scegliendo il $\max(Q(s_{t+1}, a_{t+1}))$. Questo vuole dire ipotizzare di scegliere $a_{\max} = \operatorname{argmax}(\max(Q(s_{t+1}, a_{t+1})))$, che in questo caso: $a_{\max} \neq a^*$.
Ma poi devo ripartire da capo perchè da lì in poi seleziono una sequenza diversa di transizioni di stato.



Analisi grafica delle mosse ϵ -esplorative



A.A. 2009-2010

19/25

<http://homes.dsi.unimi.it/~borghese/>



New strategy for updating eligibility trace



Eligibility set to 0 for the states in which the non-greedy action was chosen.

First decay or set to 0 the eligibility.

Then increment by 1 the eligibility of the current state.

$$e_t(s, a) = \mathcal{I}_{sst} \cdot \mathcal{I}_{aat} + \begin{cases} \gamma \lambda e_{t-1}(s, a) & \text{if } Q_{t-1}(s_t, a_t) = \max_a Q_{t-1}(s_t, a); \\ 0 & \text{otherwise,} \end{cases}$$

A.A. 2009-2010

20/25

<http://homes.dsi.unimi.it/~borghese/>



Q-learning



$$e_t(s, a) = \mathcal{I}_{ss_t} \cdot \mathcal{I}_{aa_t} + \begin{cases} \gamma \lambda e_{t-1}(s, a) & \text{if } Q_{t-1}(s_t, a_t) = \max_a Q_{t-1}(s_t, a); \\ 0 & \text{otherwise,} \end{cases}$$

Aggiorno Q:

$$Q_{t+1}(s, a) = Q_t(s, a) + \alpha \delta_t e_t(s, a),$$
$$\delta_t = r_{t+1} + \gamma \max_{a'} Q_t(s_{t+1}, a') - Q_t(s_t, a_t).$$

Scelta di a:

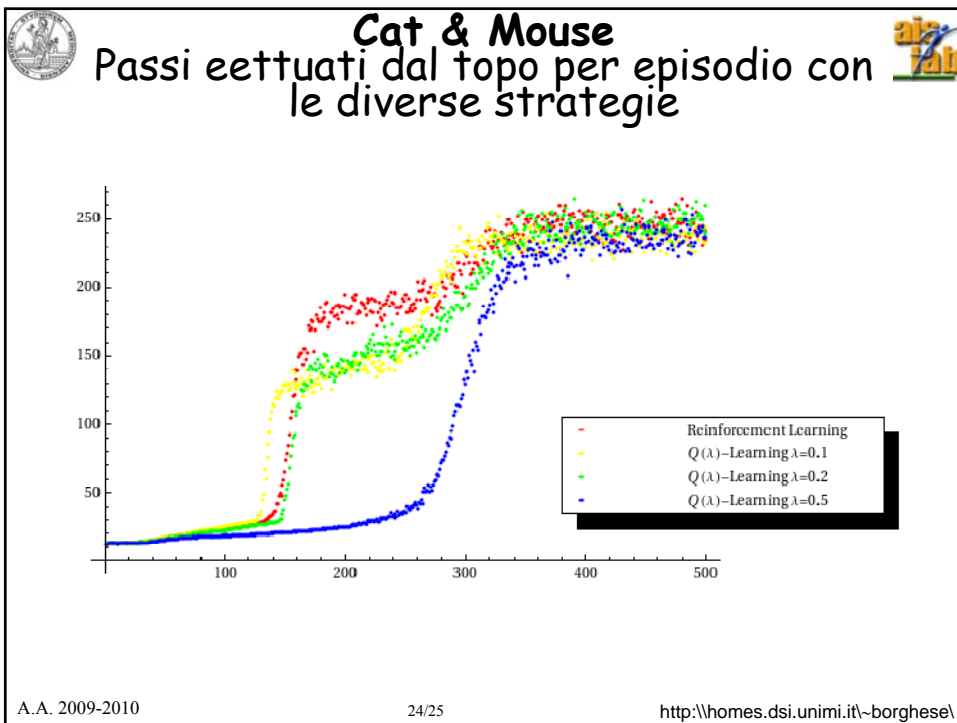
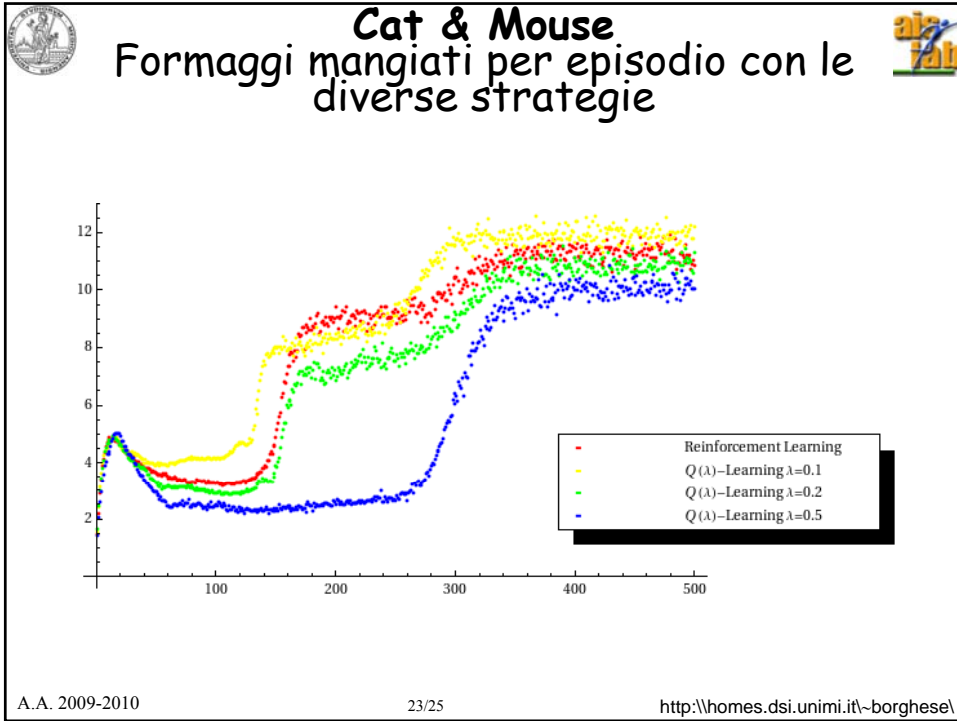
Se scelgo a_{\max} , continuo come SARSA, altrimenti $e(s, a) = 0$.



Algoritmo



Initialize $Q(s, a)$ arbitrarily and $e(s, a) = 0$, for all s, a
Repeat (for each episode):
 Initialize s, a
 Repeat (for each step of episode):
 Take action a , observe r, s'
 Choose a' from s' using policy derived from Q (e.g., ϵ -greedy)
 $a^* \leftarrow \arg \max_b Q(s', b)$ (if a' ties for the max, then $a^* \leftarrow a'$)
 $\delta \leftarrow r + \gamma Q(s', a^*) - Q(s, a)$
 $e(s, a) \leftarrow e(s, a) + 1$
 For all s, a :
 $Q(s, a) \leftarrow Q(s, a) + \alpha \delta e(s, a)$
 If $a' = a^*$, then $e(s, a) \leftarrow \gamma \lambda e(s, a)$
 else $e(s, a) \leftarrow 0$
 $s \leftarrow s'; a \leftarrow a'$
 until s is terminal





Sommario



- The eligibility trace
- $Q(\lambda)$, SARSA(λ)