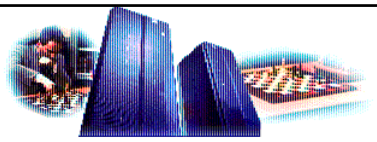
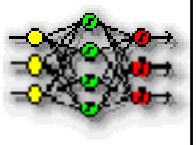




Sistemi Intelligenti Reinforcement Learning: Eligibility Trace

Alberto Borghese

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

Sommario

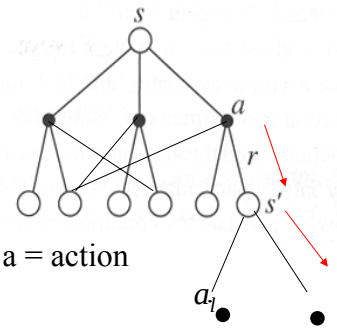
- The eligibility trace
- Q(λ), SARSA(λ)

Come apprendere Q: SARSA

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

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

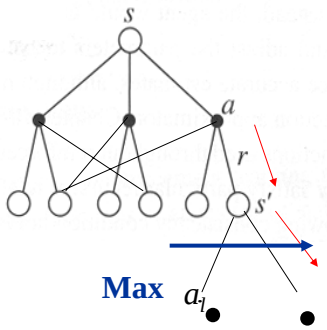
On-policy learning.

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.



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.

Formulazione di TD(0) per

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

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

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

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 assignement).

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

View of the idea – forward view

Considero il reward su orizzonti temporali più ampi:

$R_t = r_{t+1} + \gamma\, V(s_{t+1})$
$R_t = r_{t+1} + \gamma\, r_{t+2} + \gamma^2\, V(s_{t+2})$
$R_t = r_{t+1} + \gamma\, r_{t+2} + \gamma^2\, r_{t+3} + \gamma^3\, V(s_{t+3})$
……..

$$R_t = \sum_{k=0}^{M} \gamma^k r_{t+1+k} + \gamma^{k+1} V(s_{t+1+k})$$

Aggiornamento della value function: $\Delta V(s) = \alpha\, \delta_t$

$$\delta_t = R_t - V_t(s_t) = \left(\sum_{k=0}^{M} \gamma^k r_{t+1+k} + \gamma^{k+1} V(s_{t+1+k}) - V(s_t) \right)$$

Come arrivare all'elegibility trace

Considero il reward su orizzonti temporali più ampi:

$R_t = r_{t+1} + \gamma\, V(s_{t+1})$
$R_t = r_{t+1} + \gamma\, r_{t+2} + \gamma^2\, V(s_{t+2})$
$R_t = r_{t+1} + \gamma\, r_{t+2} + \gamma^2\, r_{t+3} + \gamma^3\, V(s_{t+3})$
……..

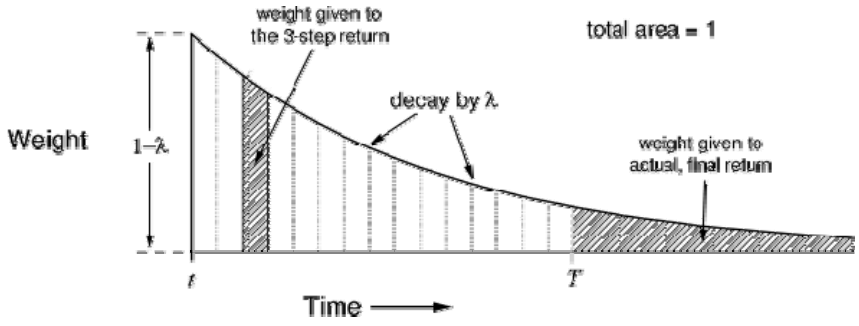
Ne faccio una media pesando di più, mediante λ, i reward nell'immediato futuro:

$R^{\lambda}_t = K * (R_t^{(1)} + \lambda\, R_t^{(2)} + \lambda^2\, R_t^{(3)} + \ldots..)$

$R^{\lambda}_t = (1-\lambda) * \sum_{k=1}^{+\infty} \lambda^{k-1} R_t^{(k)}$ Dipende da λ e da γ!

La somma dei pesi deve essere = 1

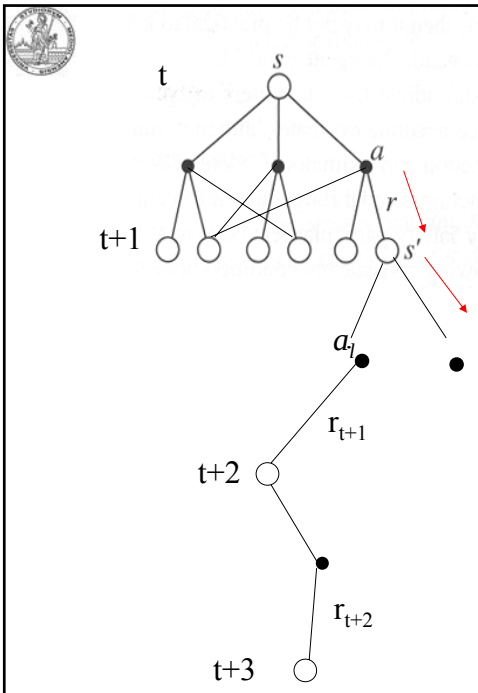
Visualizzazione grafica



λ = 0 TD(0) $\Delta V_t = \alpha\, [R_t^{\lambda} - V_t(s_t)]$

λ regola la velocità di decremento del peso del reward.

Problemi: TD(λ) in questa forma è non causale.

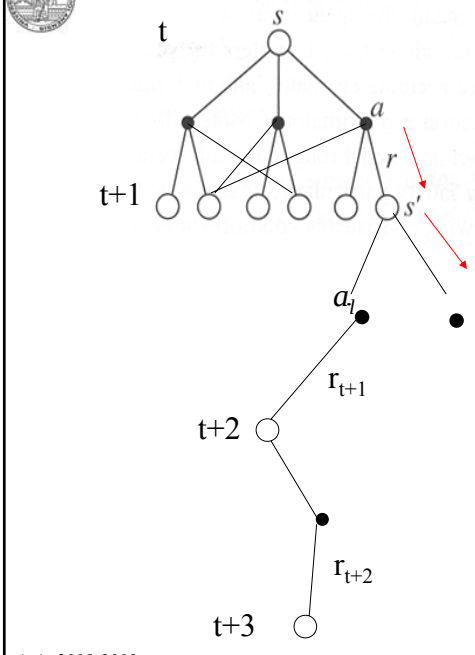


Rewards

Look forwards 3 steps
Look forwards 2 steps
Look forwards 1 step

Compute reward

Update V(s)



Rewards

We are in s_{t+3},
what shall we do?

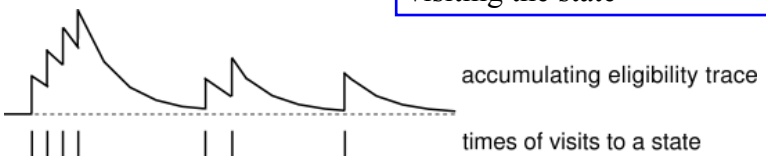
Look backwards 3 steps

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



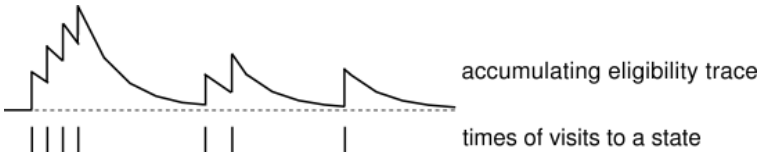
e(s) = 0 at start, e(s) ≥ 0.

Sommario

- The eligibility trace
- Q(λ), SARSA(λ)

Eligibility trace per la funzione Q

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



Come utilizzare la eligibility trace

TD(0) Learning:

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

Errore: δ_t

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

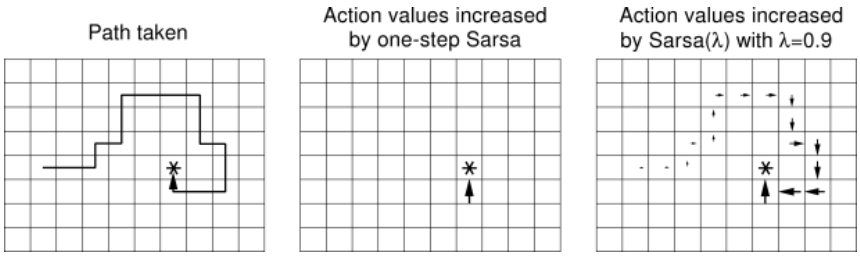
$$Q_{k+1}(s,a) = Q_k(s,a) + \alpha\delta_t e_t(s,a)$$ Per tutte le coppie (s,a)

Eleggibilità: $e_t(s,a)$

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

Esempio



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.


Watkin's Q(λ)

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

Quanto posso guardare in avanti (look-ahead)?

Ovverosia, quanto posso propagare all'indietro l'"errore"?

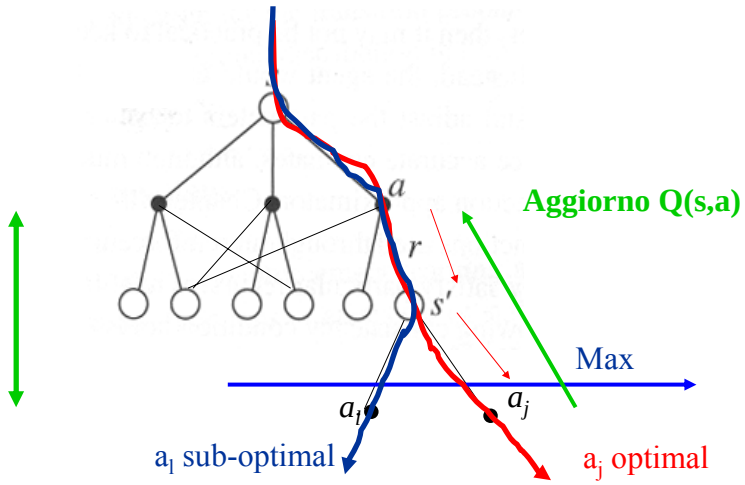

Watkin's Q(λ)

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

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} = argmax(max(Q(st+1, at+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-eplorative



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:

$$Q_{t+1}(s,a) = Q_t(s,a) + \alpha\,\delta_t\, e_t(s,a),$$

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

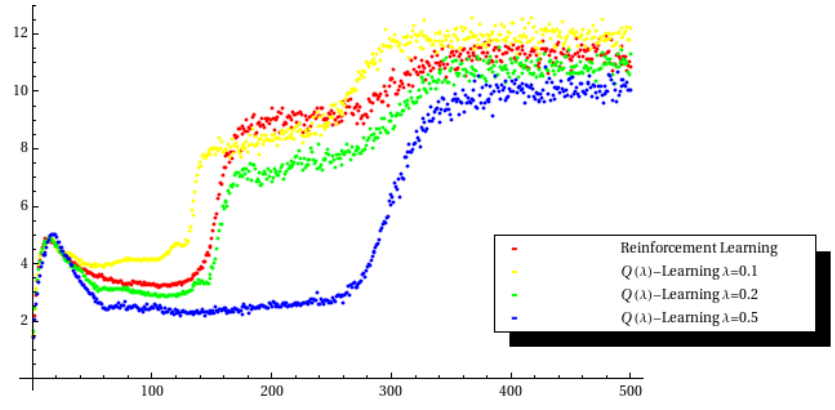
Scelta di a:

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

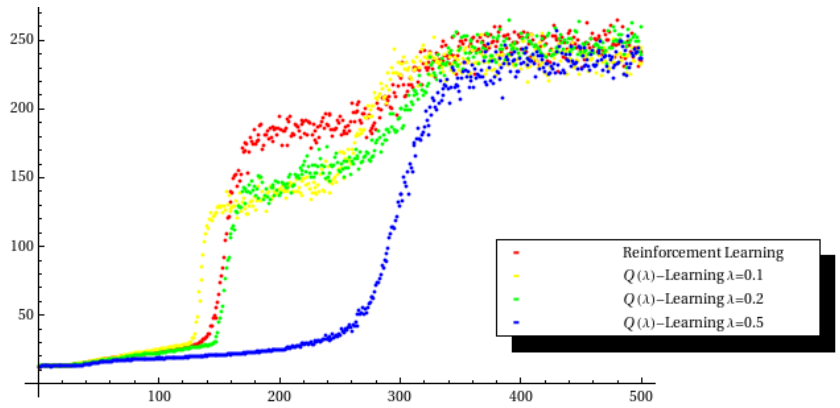
Algoritmo

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

Cat & Mouse
Formaggi mangiati per episodio con le diverse strategie



Cat & Mouse
Passi eettuati dal topo per episodio con le diverse strategie



Sommario

- The eligibility trace
- Q(λ), SARSA(λ)