

# Sistemi Intelligenti Reinforcement Learning: Policy iteration

Alberto Borghese

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



A.A. 2006-2007

1//26

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



## Sommario



**Iterative policy evaluation.**

Teorema del miglioramento della policy

Determinazione della policy ottima: policy iteration

A.A. 2006-2007

2//26

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



## Framework: ambiente + agente



### Ambiente Markoviano:

$$\Pr\{s_{t+1} = s' \mid s_t, a_t\}$$

$$\Pr\{r_{t+1} = r' \mid s_t, a_t, s_{t+1}\}$$

### Policy stocastica:

$$\pi(s, a)$$



## Framework



Descrivo il reward a lungo termine tramite la Value function:

$$V^\pi(s) = \left[ \sum_{a_j} \pi(a_j, s) \right] \sum_k P_{s \rightarrow s_k | a_j} \left[ R_{s \rightarrow s_k | a_j} + \gamma V^\pi(s_k) \right]$$

$$V^\pi(s) = E_\pi \left\{ r_{t+1} + \gamma V^\pi(s_k) \mid s_t = s \right\}$$

$$\begin{aligned} s_t &= s \\ s_{t+1} &= s_k \end{aligned}$$

$$V^\pi(s) = E_\pi \left\{ \sum_{j=0}^{+\infty} \gamma^j r_{t+j} \mid s_t = s \right\}$$



## Calcolo ricorsivo della Value function ottima



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

$V^*(s)$  di uno stato, quando viene scelta la policy ottima, deve essere uguale al valore atteso del reward per l'azione migliore per lo stato  $s$ .

$$V^*(s) = \max_{a_j} \sum_{s'} P_{s \rightarrow s' | a_j} \left[ R_{s \rightarrow s' | a_j} + \gamma V^*(s') \right]$$



## Policy Evaluation



Problemi con orizzonte temporale infinito.

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

Bellman's equation

Supponiamo di avere una policy prefissata:  $\pi(s, a)$

### Problema di policy evaluation.

Come mai posso determinare la Value function per la policy  $\pi()$ , se questa si basa sul reward che riceverò negli istanti futuri?

Sistema lineare in  $|S|$  incognite ( $|S|$  è la cardinalità dello stato).  
Metodo Montecarlo. Serve un modello dell'ambiente. Simulazione numerica.



## Iterative policy evaluation



Evoluzione del sistema da  $s(t=0)$  a  $\{s'(t = T)\}$  utilizzando la policy  $\pi(s,a)$ , prefissata.

Quanto valgono gli stati?

Parto da  $V(s(t=0))_{k=0}$  arbitraria, otterrò una value function per ogni stato che sarà funzione di  $V(s(t=0))$ .

Devo migliorare, come?

Utilizziamo l'informazione sul **passato**.



## Fondamenti del metodo

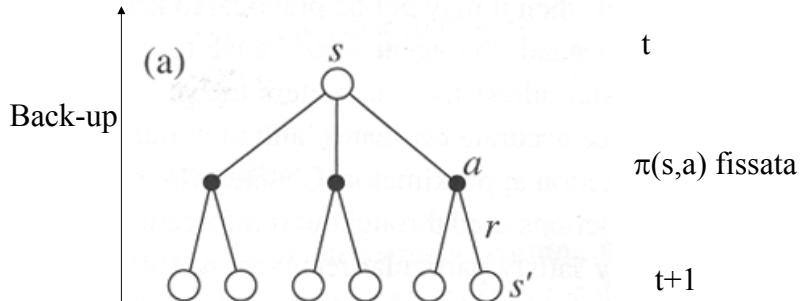


- Supponiamo di essere all'istante  $t$ . In questo istante  $t$ , si può passare ad un certo insieme di stati:  $\{s'_{t+1}\}$ .

- Analizziamo un solo passo: cosa succede nella transizione da  $t$  a  $t+1$ .



## Tecnica full-backup



Conosciamo  $V_k(s(t)) \forall s$ , anche per  $s'$  quindi  
 Analizziamo la transizione da  $s(t) \rightarrow \{s'(t+1)\}$   
 Calcoliamo un nuovo valore di  $s$ :  $V_{k+1}(s(t))$  congruente con  $V_k(s(t))$  ed  $r_{t+1}$   
 Full backup se esaminiamo tutti gli  $s'$  (cf. DP).

A.A. 2006-2007

9//26

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



## Calcolo della Value Function

Per ogni stato  $s$ , estratto a caso, analizziamo una singola transizione.

Equazione di Bellman per “iterative policy evaluation”:

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

Mi fido di  $V_k(s')$  (Backup)

$$\lim_{k \rightarrow \infty} \{V_k(s)\} = V^\pi(s)$$

A.A. 2006-2007

10//26

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



## Algoritmo per "iterative policy evaluation"



Partiamo da una politica  $\pi(s,a)$  data.

Inizializziamo  $V(s) = 0 \forall s$ , compreso gli stati finali.

Repeat

```
{
  Δ = 0;
  for s = 1 : N           // ∀s, ≠ TS
  {
    Value = V(s);
```

$$V_{k+1}(s) = \sum_{a_j} \pi(s, a_j) \sum_{s'} P_{s \rightarrow s'}^{a_j} [R_{s \rightarrow s'}^{a_j} + \gamma V(s')]$$

$$\Delta = \max(\Delta, | \text{Value}_k - V_{k+1}(s) |)$$

```
}
} Until (Δ < threshold);
```



## Interpretazione dell'update (batch o trial)



$$V_{\text{new}}(s) = \sum_{a_j} \pi(s, a_j) \sum_{s'} P_{s \rightarrow s'}^{a_j} [R_{s \rightarrow s'}^{a_j} + \gamma V(s')]$$

Al termine dell'aggiornamento dei  $V(s)$  per tutti gli stati,  $V(s) = V_{\text{new}}(s)$ . **Aggiornamento batch.**

$$V(s) = \sum_{a_j} \pi(s, a_j) \sum_{s'} P_{s \rightarrow s'}^{a_j} [R_{s \rightarrow s'}^{a_j} + \gamma V(s')]$$

Utilizzerò in parte già il nuovo valore di  $V(s)$  all'interno dell'equazione di aggiornamento. **Aggiornamento per trial.**

Entrambe le modalità di aggiornamento convergono.



## Esempio



- L'agente evolve esplorando gli stati 1-14. TS sono gli stati terminali.

- L'agente può spostarsi: {dx, sx, alto, basso}

- $\pi(s,a)$  è equiprobabile.

- $R = -1$  per ogni transizione, tranne quando  $s' = TS$  ( $R = 0$ ).

- $\gamma = 1$ .

TS	1	2	3
4	5	6	7
8	9	10	11
12	13	14	TS



## Sommario



Iterative policy evaluation.

**Teorema del miglioramento della policy**

Determinazione della policy ottima: policy iteration



## Miglioramento della policy

Tutti gli stati sono valutati in funzione di una policy data.

Condizioni di funzionamento dell'agente:

Policy deterministica:  $a = \pi(s)$ .

Ambiente stocastico.

Cosa succede se cambiamo la policy per un certo stato  $s$ ?  $a' \neq \pi(s)$ .

Cosa viene influenzato?



## Effetto del cambiamento della policy

Cambia,  $a$ , cambiano i possibili stati successivi ad  $s$ :

$$Q^\pi(s, a_{new}) = E_\pi \{ r_{t+1} + \gamma V^\pi(s_{t+1}) \mid s_t = s, a_t = a_{new} \neq \pi(s) \} =$$

$$\sum_{s'} P_{s \rightarrow s'}^{a_{new}} [R_{s \rightarrow s'}^{a_{new}} + \gamma V^\pi(s')] ]$$

?

$$Q^\pi(s, a_{new}) \geq Q^\pi(s, a = \pi(s)) = V^\pi(s)$$





## Enunciato del teorema del miglioramento della policy

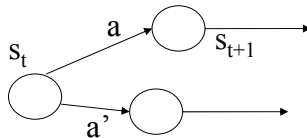


$$Q^\pi(s, a) = \sum_k P_{s \rightarrow s_k | a} [R_{s \rightarrow s_k | a} + \gamma V^\pi(s_k)]$$

**Ipotesi:**  $Q^\pi(s, \pi'(s)) \geq V^\pi(s)$

$$Q^\pi(s, a_{new} = \pi'(s, a)) = \sum_k P_{s \rightarrow s_k | a_{new}} [R_{s \rightarrow s_k | a_{new}} + \gamma V^\pi(s_k)]$$

**Tesi:**  $\pi'$  è meglio di  $\pi$ . Cioè:  $V^{\pi'}(s) \geq V^\pi(s) \forall s$ .



A.A. 2006-2007

17/26

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



## Dimostrazione del teorema del miglioramento della policy



**Analizziamo la seguente condizione:**

$\pi' = \pi \forall s$  tranne che per  $s_m$  per il quale si applica l'azione:  
 $a_{new} = \pi'(s_m)$

Risulta che  $a$  è migliore di  $a_{new} = \pi(s)$ .

$$Q^{\pi'}(s, a_{new} = \pi'(s)) \geq Q^\pi(s, a = \pi(s)) = V^\pi(s)$$

**Tesi:**  $\pi'$  è meglio di  $\pi$ . Cioè:  $V^{\pi'}(s) \geq V^\pi(s) \forall s$ .

A.A. 2006-2007

18/26

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



## Dimostrazione del teorema del miglioramento della policy



Hp:  $Q^\pi(s, \pi'(s)) \geq V^\pi(s) \quad \forall s \quad \pi'(s, a)$  è migliore per almeno uno stato

$$V^\pi(s) \leq Q^\pi(s, \pi'(s))$$

$$= E_{\pi'}\{r_{t+1} + \gamma V^\pi(s_{t+1}) \mid s_t = s\}$$

$$\leq E_{\pi'}\{r_{t+1} + \gamma Q^\pi(s_{t+1}, \pi'(s_{t+1})) \mid s_t = s\}$$

$$\leq E_{\pi'}\{r_{t+1} + \gamma E_{\pi'}(r_{t+2} + \gamma V^\pi(s_{t+2})) \mid s_t = s\}$$

$$= E_{\pi'}\{r_{t+1} + \gamma r_{t+2} + \gamma^2 V^\pi(s_{t+2}) \mid s_t = s\}$$

Sostituisco ancora  $Q^{\pi^*}(\cdot)$

$$\leq E_{\pi'}\{r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \dots \mid s_t = s\}$$

$$\text{Th: } V^\pi(s) \leq V^{\pi'}(s)$$



## Sommario



Iterative policy evaluation

Teorema del miglioramento della policy

**Determinazione della policy ottima: policy iteration**



## Politica ottima

Miglioramento della politica per tutti gli stati.

$$\begin{aligned}
\pi'(s_k) &= \arg \max_a Q^\pi(s, a) && \text{greedy o } \varepsilon\text{-greedy} \\
\forall s &= \arg \max_a E\{r_{t+1} + \gamma V^\pi(s') \mid s_t = s, a_t = a\} \\
&= \arg \max_a \sum_{s'} P_{s \rightarrow s'}^a [R_{s \rightarrow s'}^a + \gamma V^\pi(s')]
\end{aligned}$$

### Policy improvement

Si può estendere al caso di comportamento stocastico dell'agente nel qual caso:  $\pi(s,a)$  è una probabilità.



## Policy ottima ed equazioni di Bellman

Se sono in questa condizione (politica ottimale)

$$\pi'(s_k) = \arg \max_a Q^\pi(s, a)$$

Se abbiamo la corrispondente value function,  $V(s)$ ,

Allora vale la relazione di Bellman tra le value function ottime:

$$V^*(s) = \max_{a_j} \sum_{s'} P_{s \rightarrow s'|a_j} [R_{s \rightarrow s'|a_j} + \gamma V^*(s')]$$



# Policy iteration

Iterazione tra:

- Calcolo iterativo della Value function (iterative policy evaluation)
- Miglioramento della policy (policy improvement)

$$\pi_0 \rightarrow V^{\pi_0} \rightarrow \pi_1 \rightarrow V^{\pi_1} \rightarrow \pi_2 \rightarrow V^{\pi_2} \rightarrow \dots$$

$$\rightarrow \pi^* \rightarrow V^*$$

Converge velocemente ad una buona politica



# Algoritmo (progetto per esame)

Repeat until  
policy-stable

## 1. Initialization

$V(s) \in \mathfrak{R}$  and  $\pi(s) \in \mathcal{A}(s)$  arbitrarily for all  $s \in \mathcal{S}$

## 2. Policy Evaluation

Repeat

$\Delta \leftarrow 0$

For each  $s \in \mathcal{S}$ :

$v \leftarrow V(s)$

$V(s) \leftarrow \sum_{s'} \mathcal{P}_{ss'}^{\pi(s)} [\mathcal{R}_{ss'}^{\pi(s)} + \gamma V(s')]$

$\Delta \leftarrow \max(\Delta, |v - V(s)|)$

until  $\Delta < \theta$  (a small positive number)

## 3. Policy Improvement

*policy-stable*  $\leftarrow$  true

For each  $s \in \mathcal{S}$ :

$b \leftarrow \pi(s)$

$\pi(s) \leftarrow \arg \max_a \sum_{s'} \mathcal{P}_{ss'}^a [\mathcal{R}_{ss'}^a + \gamma V(s')]$

If  $b \neq \pi(s)$ , then *policy-stable*  $\leftarrow$  false

If *policy-stable*, then stop; else go to 2



## Esercizio: autonoleggio

2 locazioni di autonoleggio

In ogni locazione, se quando arriva un cliente l'auto è disponibile, si guadagnano 10 euro.

Se l'auto non è disponibile si perde la gestione della locazione.

Le auto diventano disponibili il giorno dopo che sono state restituite dopo il noleggio.

Si possono portare le auto da un autonoleggio all'altro di notte al costo di 2 Euro per ogni auto.

Supponiamo che il numero di auto richieste e restituite in ognuno dei 2 autonoleggi sia rappresentato da una distribuzione di Poisson (probabilità che vengano richieste  $n$  auto:  $(\lambda^n / n!)e^{-\lambda}$  dove  $\lambda$  è il valore atteso (media) di auto richieste o restituite).

Supponiamo che non ci possano essere più di 20 auto in uno dei 2 autonoleggi (ciascuna auto aggiuntiva viene inviata al centro di raccolta nazionale della compagnia).

Supponiamo anche che un massimo di 5 auto possa essere spostato in una singola notte.

Questo problema si può formulare con un MDP dove lo stato è rappresentato dal numero di auto presenti in ciascuno dei 2 autonoleggi al termine di una giornata e le azioni il numero di auto che vengono spostate durante la notte.

Consideriamo  $\gamma = 0.9$ . Il reward sarà il reward accumulato durante il giorno + notte.

**Partiamo dalla policy  $\pi(s,a) = 0$ : nessuna auto viene mossa. Siamo in grado di migliorarla? Come?**

A.A. 2006-2007

25/26

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



## Sommario

Iterative policy evaluation.

Teorema del miglioramento della policy

Determinazione della policy ottima: policy iteration

A.A. 2006-2007

26/26

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