

Sistemi Intelligenti Reinforcement Learning: Processi Markoviani e Value Function

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. 2005-2006

1/29

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



Sommario



Il Reinforcement Learning.

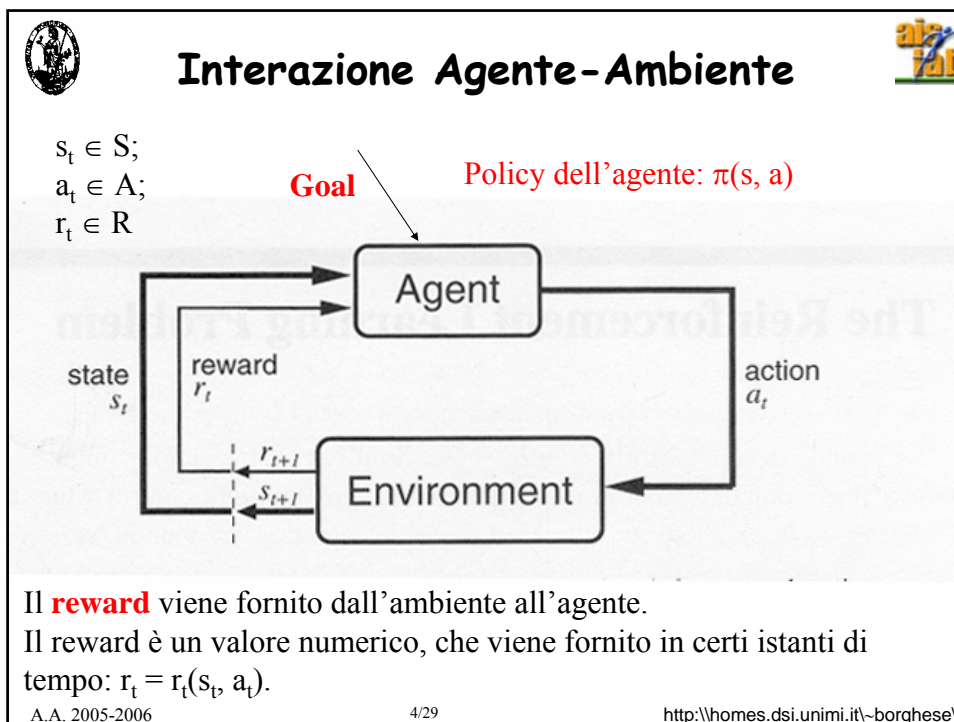
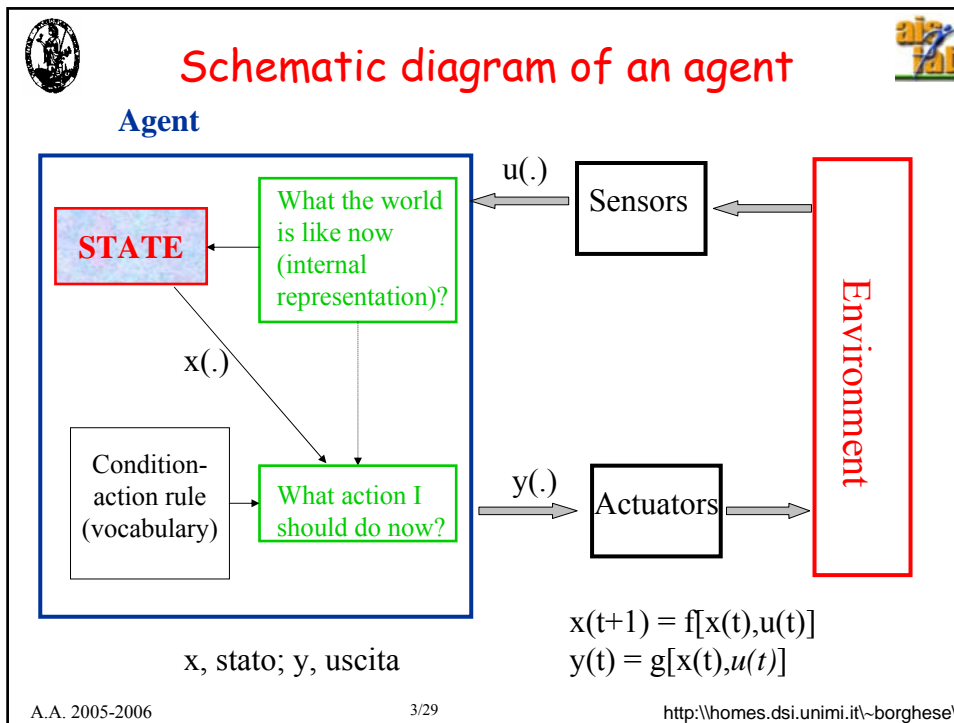
Processi Markoviani.

La value function: ricompensa a lungo termine: formulazione ricorsiva.

A.A. 2005-2006

2/29

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

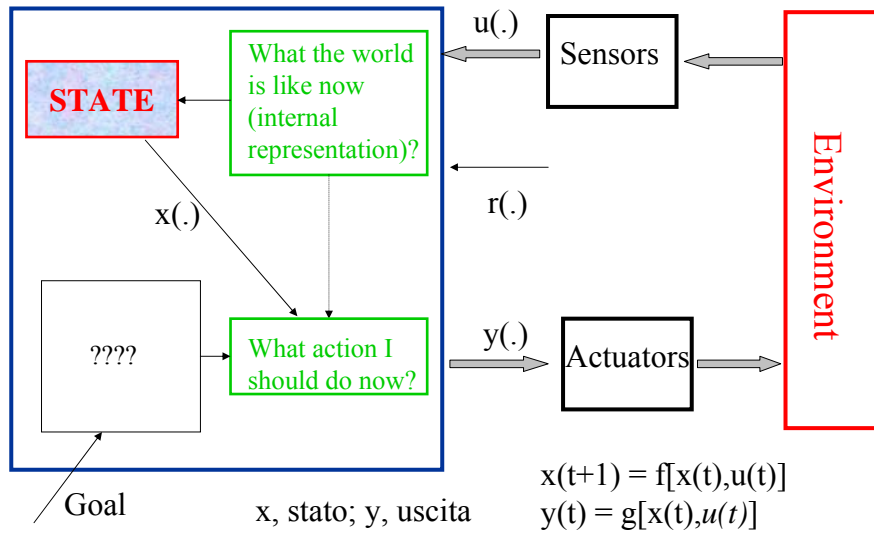




Schematic diagram of an agent in RL



Agent



A.A. 2005-2006

5/29

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



Osservazioni



Formulazione generale che si adatta ad una grande quantità di problemi.

Agente = Controllore.

Tempo = tempo, ma anche stadio della decisione, del planning....

Azione = forza, voltaggio, decisioni.....

Stato = situazione = misura di grandezze fisiche, di grandezze interne, stato mentale,....

Pre-processing di misure fisiche. E' importante per un efficiente RL.

Ambiente = tutto quanto non è modificabile direttamente dall'agente. Può essere noto o meno.

Reward = viene generato all'esterno dell'agente.

Value = viene stimata all'interno dell'agente.

A.A. 2005-2006

6/29

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



Caratteristiche dello stato



Policy di un agente: $\pi(s, a)$.

Caratteristiche dello stato, s :

- Contiene gli stimoli pre-elaborati a partire dagli stimoli semplici misurati sull'ambiente.
- Gli stimoli pre-elaborati analizzano una sequenza temporale di stimoli semplici.
- Lo stato deve potere essere misurato dall'agente.

Come rappresento s ? Memorizzo la sequenza temporale degli stimoli semplici di interesse?



Reward e Obiettivi



Il reward è "esterno" all'agente.

Massimizzare la ricompensa a lungo termine, Value, cumulando le ricompense istantanee: $r_t(a(t)) \in \mathbb{R}$.

Definendo una ricompensa che viene massimizzata solamente quando il goal viene raggiunto, possiamo ottenere che l'agente impari il task (raggiunga il goal).

Collegamento tra reward e goal.

Il reward consente di comunicare COSA si vuole ottenere; nulla è detto sul COME.



Value function



Cosa si intende per ricompensa a lungo termine?
Questa è rappresentata dalla **Value Function**; cosa rappresenta?

Al tempo t , data una certa policy: $\pi(s, a)$, la ricompensa sarà una funzione dei reward negli istanti di tempo successivi a t , ad esempio:

$$R_t = r_{t+1} + r_{t+2} + r_{t+3} + \dots + r_T$$

Quando è adeguata?
Problemi ad orizzonte finito (episodic tasks, a terminal state is defined).
Problemi stazionari.



Infinite horizon problems (continuing tasks)



Il concetto fondamentale è il “**discount**”.

Discounted reward o discounted return:

$$R_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+2} + \dots = \sum_{k=0}^{+\infty} \gamma^k r_{t+(k+1)}$$

Dove $0 \leq \gamma \leq 1$ è il “discount rate”.



Sommario



Il Reinforcement Learning.

Processi Markoviani.

La value function: ricompensa a lungo termine: formulazione ricorsiva.



Environment Markoviano



Una variabile di stato (non funzione del tempo), che riassume le informazioni sulla storia del task, utili all'agente per agire, è detta variabile Markoviana.

Formalizziamo. Supponiamo s ed r variabili discrete appartenenti ad un insieme finito di valori.

$$\Pr\{s_{t+1} = s' \mid s_t, a_t, r_t; s_{t-1}, a_{t-1}, r_{t-1}; \dots; s_0, a_0, r_0\}$$

Se lo stato è Markoviano:

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



Reinforcement Markoviano



Reward stocastico.

$$\Pr\{r_{t+1} = r' \mid s_t, a_t, r_t; s_{t-1}, a_{t-1}, r_{t-1}; \dots; s_0, a_0, r_0\}$$

Se lo stato è Markoviano:

$$\text{Reward stocastico: } \Pr\{s_{t+1} = s'; r_{t+1} = r' \mid s_t, a_t\}$$

$$\text{Reward deterministico: } r_{t+1} = r(s_t, a_t).$$

L'ambiente ha completamente proprietà Markoviane.

I modelli Markoviani sono modelli molte generali!



Approssimazione



L'ipotesi di ambiente e rinforzo Markoviani possono essere soddisfatte in modo approssimato.

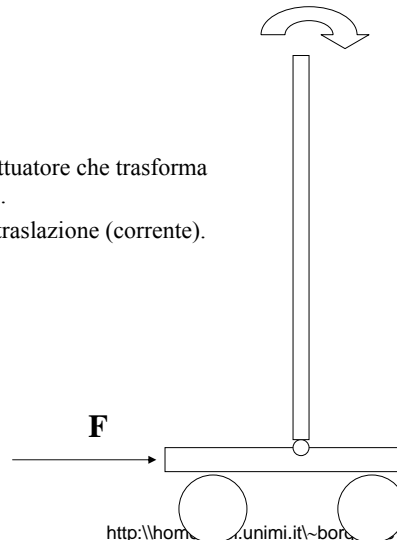
Environment – cart + pole.

Stato – Posizione e velocità di cart e di pole, attuatore che trasforma la corrente per il motore in forza di traslazione.

Agente – Controllore del motore che pilota la traslazione (corrente).

Reward – Cade / non_cade - 0 / 1.

Value Function ?





Postural control is a complex problem



Complex system: multi-input – multi-output (each leg has 56 major muscle groups).

It is a non-linear system. High coupling between body segments (e.g. biarticular muscles).

Muscles bandwidth is limited.

The control system introduces delays, increasing from the periphery to the CNS.

Classical control theory is “difficult”.

Nevertheless, we learn upright posture in the very first year of our life.



Markov decision process



(Finite) Markov Decision Process.

$$P_{s \rightarrow s' | a} = \Pr \{s_{t+1} = s' | s_t = s, a_t = a\}$$

$$R_{s \rightarrow s' | a} = E \{r_{t+1} = r' | s_t = s, a_t = a, s_{t+1} = s'\}$$

Descrizione dell'ambiente



Esempio: AIBO search



Azioni:

- 1) Rimanere fermo e aspettare che qualcuno getti nel cestino una lattina vuota.
- 2) Muoversi attivamente in cerca di lattine.
- 3) Tornare alla sua base (recharge station) e ricaricarsi.

Stato:

- 1) Alto livello di energia.
- 2) Basso livello di energia.

Policy:

$A(s = \text{high}) = \{\text{Search, Wait}\}$

$A(s = \text{low}) = \{\text{Search, Wait, Recharge}\}$

A.A. 2005-2006

17/29

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



Funzionamento del Robot



Funzione Stato prossimo:

$$P_{s \rightarrow s' | a} = \Pr\{s_{t+1} = s' | s_t = s, a_t = a\}$$

Se il livello di energia è alto ($s_t = \text{alto}$):

se scelgo Wait - $s_{t+1} = \text{alto}$.

se scelgo Search, s_{t+1} avrà una certa probabilità di diventare low.

$$P_{\text{high} \rightarrow \text{low} | \text{Search}} = \Pr\{s_{t+1} = \text{low} | s_t = \text{high}, a_t = \text{Search}\} = \alpha$$

Se il livello di energia è basso ($s_t = \text{basso}$):

se scelgo Wait - $s_{t+1} = \text{basso}$.

se scelgo Recharge - $s_{t+1} = \text{alto}$.

se scelgo Search, s_{t+1} avrà una certa probabilità di fermarsi.

$$P_{\text{low} \rightarrow \text{low} | \text{Search}} = \Pr\{s_{t+1} = \text{low} | s_t = \text{low}, a_t = \text{Search}\} = \beta$$

A.A. 2005-2006

18/29

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



Reward del Robot



$$R_{s \rightarrow s'|a} = E\{r_{t+1} = r' | s_t = s, a_t = a, s_{t+1} = s'\}$$

Funzione Reward:

R^{search} reward se il robot sta cercando.

R^{wait} reward se il robot sta aspettando.

$R^{\text{search}} > R^{\text{wait}}$.

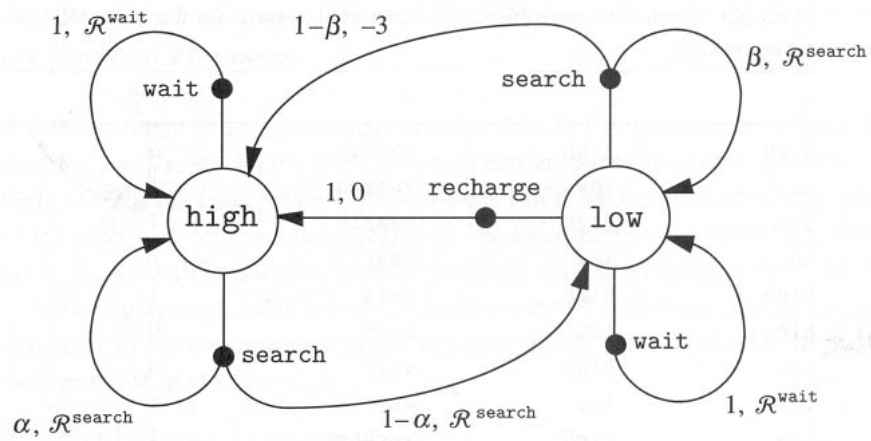


Forma tabellare

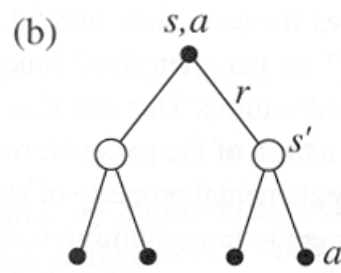
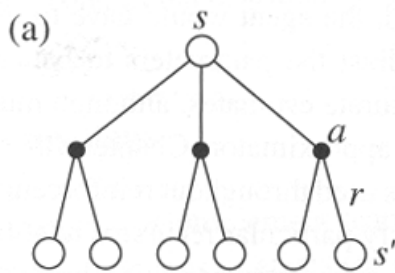
s	s'	a	$P_{s \rightarrow s' a}$	$R_{s \rightarrow s' a}$
alta	alta	ricerca	α	R^{search}
alta	bassa	ricerca	$1 - \alpha$	R^{search}
bassa	alta	ricerca	$1 - \beta$	-3
bassa	bassa	ricerca	β	R^{search}
alta	alta	attesa	1	R^{wait}
alta	bassa	attesa	0	R^{wait}
bassa	alta	attesa	0	R^{wait}
bassa	bassa	attesa	1	R^{wait}
bassa	alta	ricarica	1	0
bassa	bassa	ricarica	0	0
alta	Non esiste	ricarica	X	X



Transition graph



Policy



La policy deve essere determinata.



Sommario



Il Reinforcement Learning.

Processi Markoviani.

La value function: ricompensa a lungo termine: formulazione ricorsiva.



La value function



Nulla è detto sulla policy: dato uno stato, in quale nodo azione mi sposto?

Vogliamo costruire agenti lungimiranti.

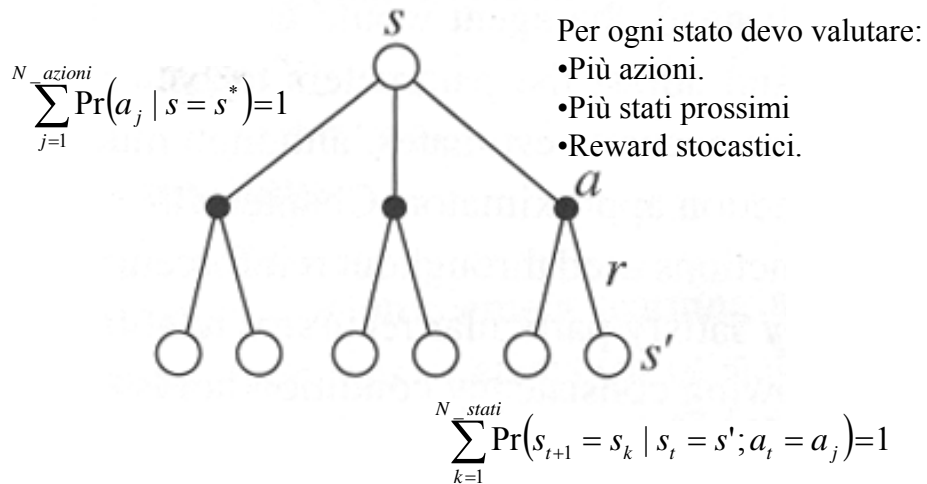
State-Value function

$$V^\pi(s) = E_\pi \{R_t \mid s_t = s\} = E_\pi \left\{ \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid s_t = s \right\}$$

Massimizzo la ricompensa a lungo termine, $V(\cdot)$. Dipende dalla policy:



Value function e modelli markoviani



A.A. 2005-2006

25/29

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



La Value function sulle azioni



La value function può riguardare le azioni.

Action-Value function

$$Q^\pi(s, a) = E_\pi \{R_t | s_t = s, a_t = a\} = E_\pi \left\{ \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid s_t = s, a_t = a \right\}$$

Massimizzo la ricompensa a lungo termine, $Q(\cdot)$. Dipende dalla policy.

Dove abbiamo già trovato una Value function associata alle azioni?

A.A. 2005-2006

26/29

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



Reinforcement Learning Problem



Given: Repeatedly...

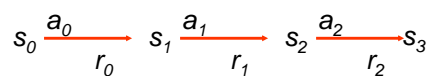
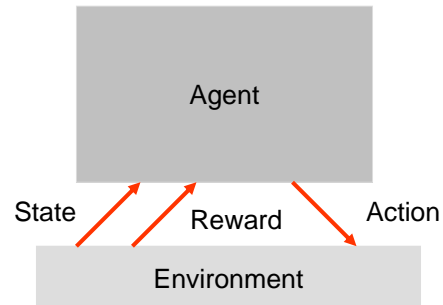
- Executed action
- Observed state
- Observed reward

Learn action policy $\pi: S \rightarrow A$

- ◆ Maximizes life reward
 $r_0 + \gamma r_1 + \gamma^2 r_2 \dots$
from any start state.
- ◆ Discount: $0 < \gamma < 1$

Note:

- Unsupervised learning
- Delayed reward



Goal: **Learn** to choose actions that maximize life reward

$$r_0 + \gamma r_1 + \gamma^2 r_2 \dots$$



How About Learning the Policy Directly?



1. $\pi^*: S \rightarrow A$
2. fill out table entries for π^* by collecting statistics on training pairs $\langle s, a^* \rangle$.
3. Where does a^* come from?

Problema di ottimizzazione della Value function, basato sulla sua stima intelligente.



Sommario



Il Reinforcement Learning.

Processi Markoviani.

La value function: ricompensa a lungo termine: formulazione ricorsiva.