

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. 2007-2008

1/26

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



Sommario



Il Reinforcement Learning.

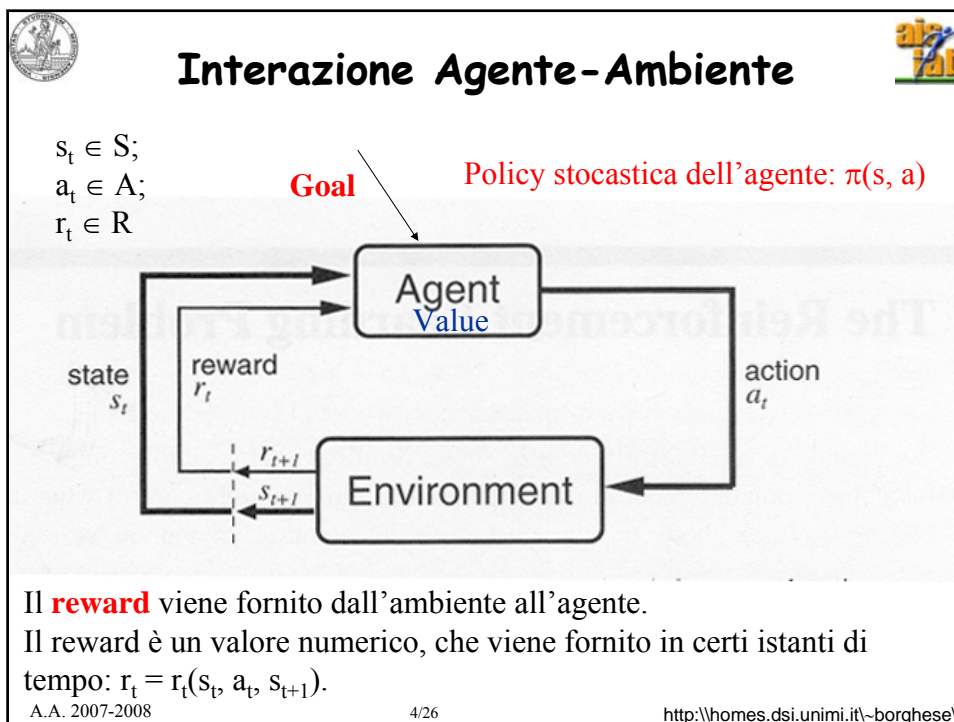
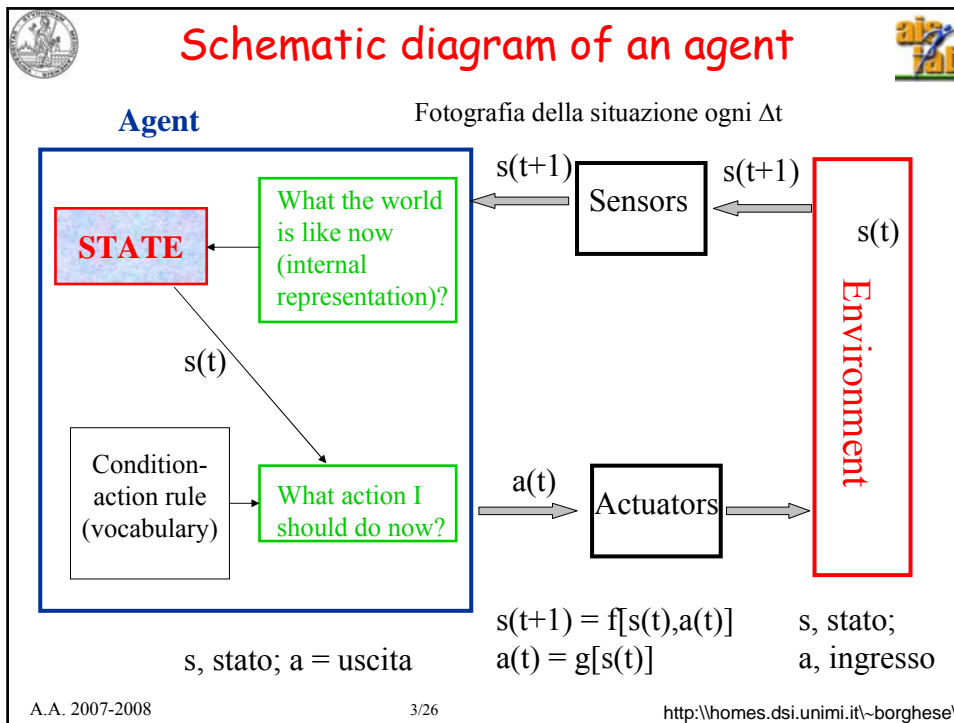
Processi Markoviani.

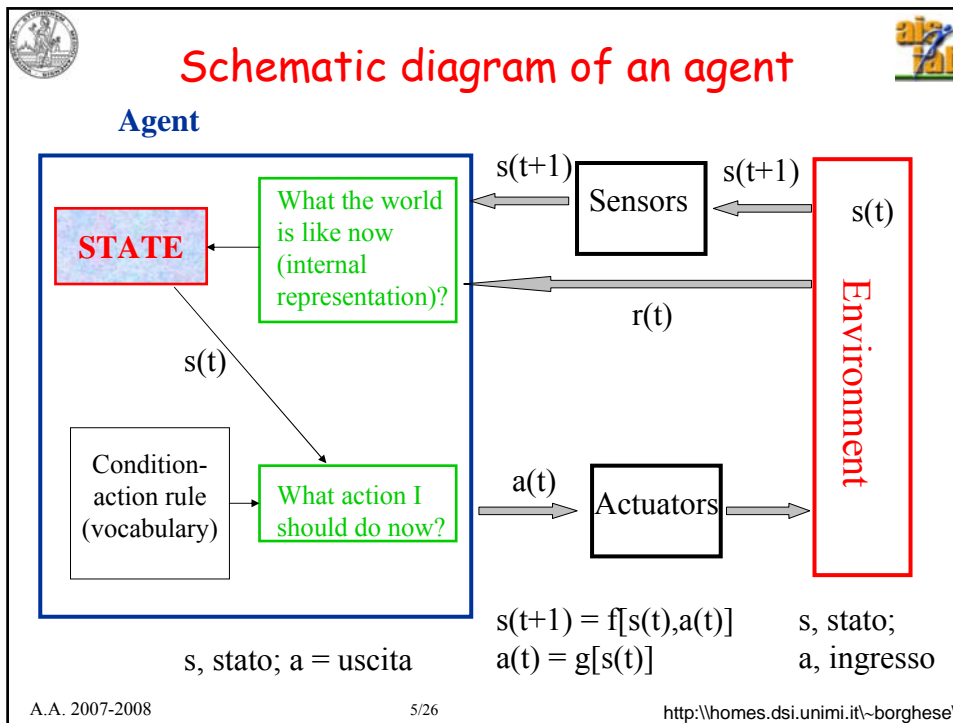
La value function: ricompensa a lungo termine: formulazione ricorsiva.

A.A. 2007-2008

2/26

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





Gli elementi del RL

Goal. Obiettivo che deve raggiungere l'agente. Si può aggiungere che l'agente deve raggiungere l'obiettivo con una policy ottima.

Policy. Descrive l'azione scelta dall'agente: mapping tra **stato** (output dell'ambiente) e azioni dell'agente. Funzione di controllo. Le policy possono avere una componente stocastica. Viene utilizzata una modalità adeguata per rappresentare il comportamento dell'agente (e.g. tabella, funzione continua parametrica...).

Reward function. Ricompensa **immediata**. Associata all'azione intrapresa in un certo stato. Può essere data al raggiungimento di un goal (esempio: successo / fallimento). E' uno scalare. Rinforzo primario, solitamente **qualitativo**.

Value function. "Cost-to-go". Ricompensa a **lungo termine**. Somma dei reward: costi associati alle azioni scelte istante per istante + costo associato allo stato finale.
Orizzonte temporale ampio. Rinforzo secondario.

Environment. Fornisce la reward function, fornisce l'input in base al quale l'agente aggiorna lo stato. Reagisce agli output dell'agente.

A.A. 2007-2008 6/26 http://homes.dsi.unimi.it/~borghese/



Meccanismo di apprendimento nel RL



Ciclo dell'agente (le tre fasi sono sequenziali):

- 1) Implemento una policy
- 2) Aggiorno la Value function
- 3) Aggiorno la policy.



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.



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.
- Lo stato deve essere efficiente per il raggiungimento del goal.

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:

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

Terminal State

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:

$$R_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \dots = \sum_{k=0}^{+\infty} \gamma^k r_{t+(k+1)}$$

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

Present value of future rewards.

$$R_t \rightarrow \frac{r}{1-\gamma} \quad \text{if } r_t = r_{t+k} \quad \forall k$$

Relazione con il caso non-stazionario nel setting non-associativo?

Cosa succede se $\gamma \rightarrow 0$ e $\gamma \rightarrow 1$?



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.

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

Se lo stato è Markoviano:

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

NB: Non sempre $\Pr\{s_{t+1} = s' \mid s_t, a_t\}$ è nota!



Reinforcement Markoviano



Reward stocastico.

$$\Pr\{r_{t+1} = r' \mid s_t, a_t, s_{t+1}; s_{t-1}, a_{t-1}, r_t; \dots; s_0, a_0, r_1\}$$

Se lo stato è Markoviano:

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

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

L'ambiente ha completamente proprietà Markoviane.

I modelli Markoviani sono modelli molto generali!



Markov decision process



(Finite) Markov Decision Process.

$$P_{s \rightarrow s' | a} = \Pr\{s_{t+1} = s' \mid s_t = s, a_t = a\} \quad \text{Probabilità di transizione}$$

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

Descrizione della dinamica dell'ambiente

Azione che dall'esterno viene imposta all'ambiente



Approssimazione

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

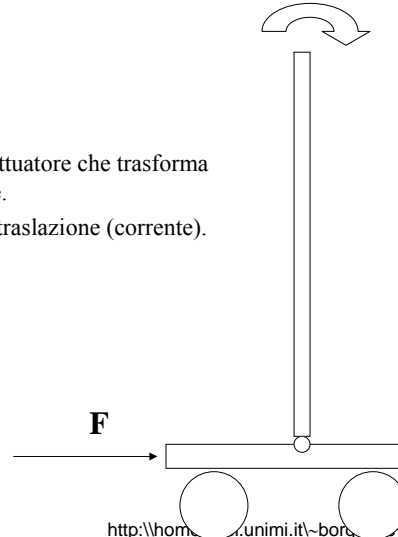
Environment – cart + pole.

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 ?



A.A. 2007-2008

17/26

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



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.



A.A. 2007-2008

18/26

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



Sommario



Il Reinforcement Learning.

Processi Markoviani.

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

A.A. 2007-2008

19/26

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



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.

Goal: collezionare il maggior numero di lattine.

Policy:

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

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

A.A. 2007-2008

20/26

<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. 2007-2008

21/26

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



Reward del Robot



Funzione Reward:

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

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

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

-3 se occorre portarlo a ricaricarsi.

0 se il robot va autonomamente a ricaricarsi.

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

A.A. 2007-2008

22/26

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



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

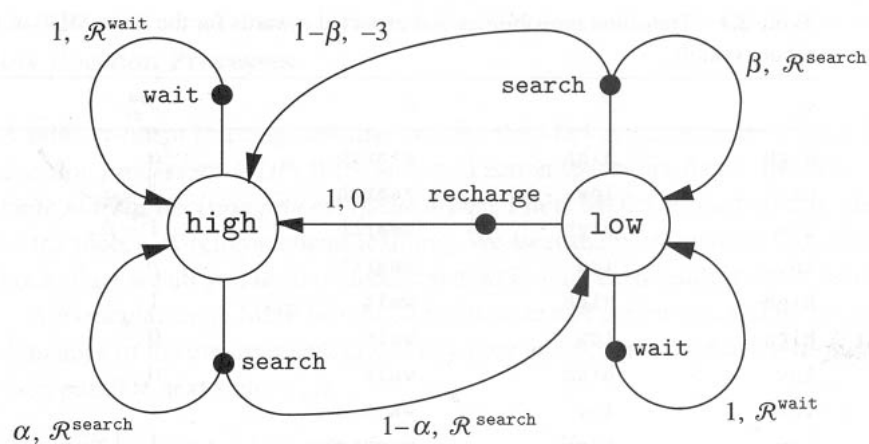
A.A. 2007-2008

23/26

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



State Transition Graph



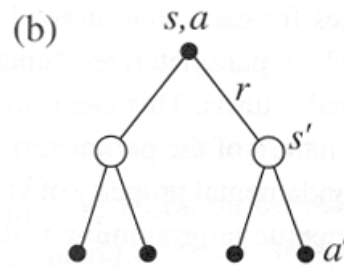
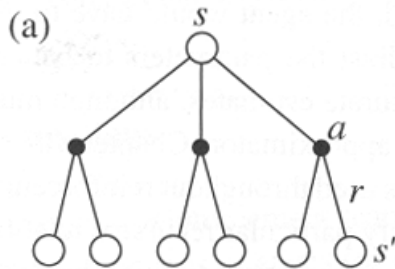
A.A. 2007-2008

24/26

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



Policy



La policy deve essere ancora determinata. Come fa l'agente a determinare la policy ottimale?

Archi multipli fuoriuscenti da un'azione sono associati alla probabilità di scegliere quel cammino (ambiente stocastico).

Archi multipli fuoriuscenti da uno stato, sono associati alla policy.



Sommario



Il Reinforcement Learning.

Processi Markoviani.

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