

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.

A.A. 2007-2008 7/26 <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. 2007-2008 8/26 <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?

A.A. 2007-2008 9/26 <http://homes.dsi.unimi.it/~borghese/>

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.

A.A. 2007-2008 10/26 <http://homes.dsi.unimi.it/~borghese/>

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.

A.A. 2007-2008 11/26 <http://homes.dsi.unimi.it/~borghese/>

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.

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

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

A.A. 2007-2008 12/26 <http://homes.dsi.unimi.it/~borghese/>

Sommaro

Il Reinforcement Learning.

Processi Markoviani.

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

A.A. 2007-2008 13/26 <http://homes.dsi.unimi.it/~borghese/>

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:

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

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

A.A. 2007-2008 14/26 <http://homes.dsi.unimi.it/~borghese/>

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:

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

L'ambiente ha completamente proprietà Markoviane.

I modelli Markoviani sono modelli molto generali!

A.A. 2007-2008 15/26 <http://homes.dsi.unimi.it/~borghese/>

Markov decision process

(Finite) Markov Decision Process.

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

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

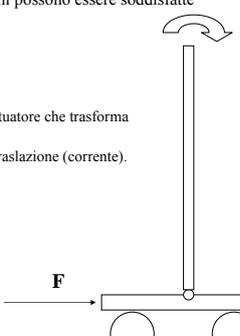
Descrizione della dinamica dell'ambiente
Azione che dall'esterno viene imposta all'ambiente

A.A. 2007-2008 16/26 <http://homes.dsi.unimi.it/~borghese/>

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/~borghese/>

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/~borghese/>

Sommaro

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.

Policy:

A(s = high) = {Search, Wait}
 A(s = low) = {Search, Wait, Recharge}

Goal: collezionare il maggior numero di lattine.

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.

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

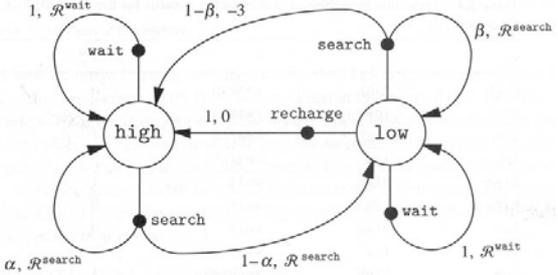
A.A. 2007-2008 22/26 <http://homes.dsi.unimi.it/~borghese/>

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



The graph shows two states: 'high' and 'low'. Transitions are labeled with (action, probability, reward):

- high to high: wait (1, R^{wait})
- high to low: search (α , R^{search})
- low to high: search ($1 - \alpha$, R^{search})
- low to low: wait (1, R^{wait})
- low to high: recharge (1, 0)
- low to low: search (β , R^{search})
- low to high: search ($1 - \beta$, -3)

A.A. 2007-2008 24/26 <http://homes.dsi.unimi.it/~borghese/>

Policy

(a)

(b)

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.

A.A. 2007-2008 25/26 <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 26/26 <http://homes.dsi.unimi.it/~borghese/>