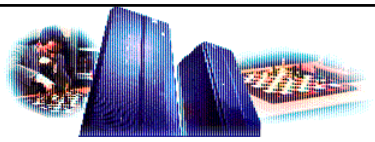
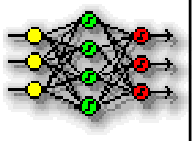




Sistemi Intelligenti
Reinforcement Learning:
Processi Markoviani e Value function

Alberto Borghese

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

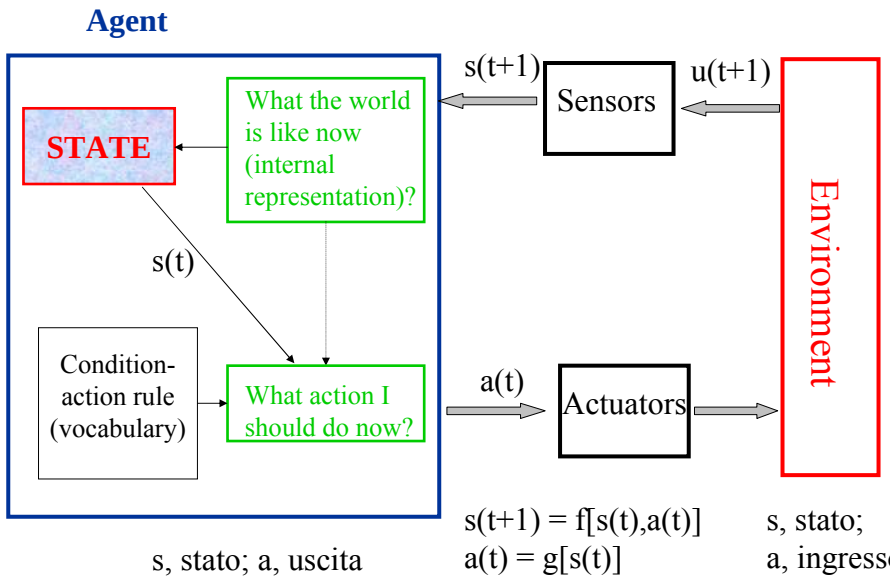
Sommario

Il Reinforcement Learning.

Processi Markoviani.

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

Schematic diagram of an agent

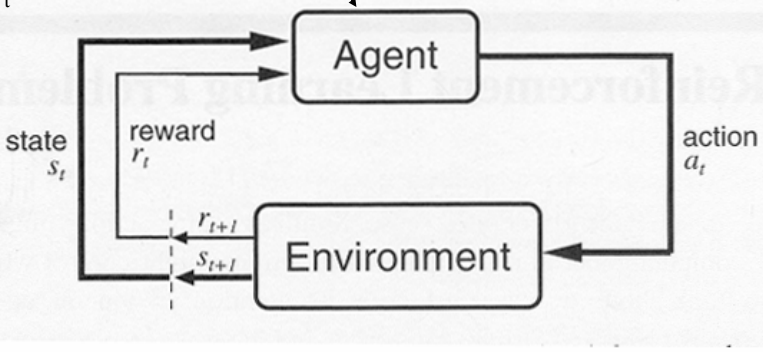


Interazione Agente-Ambiente

$s_t \in S$;
$a_t \in A$;
$r_t \in R$

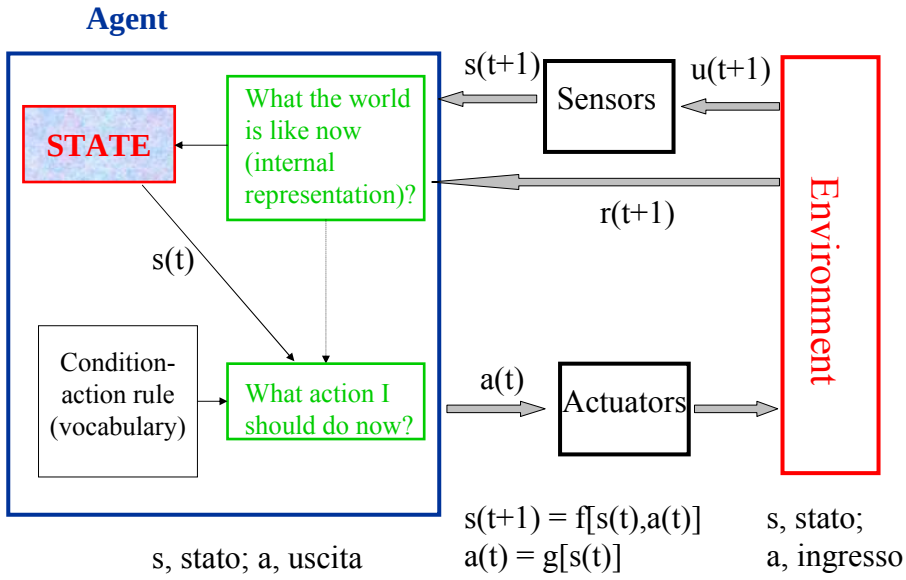
Goal

Policy dell'agente: $\pi(s, a)$



Il **reward** viene fornito dall'ambiente all'agente.
Il reward è un valore numerico, che viene fornito in certi istanti di tempo: $r_t = r_t(s_t, a_t, s_{t+1})$.

Schematic diagram of an agent



Gli attori del RL

Policy. Descrive l'azione scelta dall'agente: mapping tra stato (input dall'ambiente) e azioni. Funzione di controllo. Le policy possono avere una componente stocastica. Viene utilizzato un modello adeguato del 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.

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.

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.

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

Reward e Obbiettivi

Il reward è "esterno" all'agente.

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

Defininendo 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:

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

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:

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

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

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

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

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.

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

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!

Reinforcement Markoviano

Reward stocastico.

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

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!

Approssimazione

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

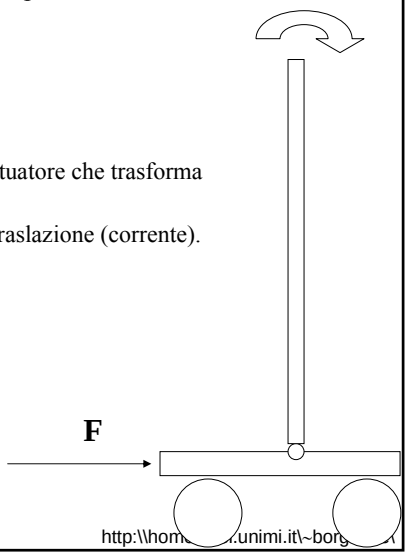
Environment – cart + pole.

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 pilata 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.

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

Probabilità di transizione

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

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

Task: collezionare il maggior numero di lattine.

Policy:

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

Funzionamento del Robot

Funzione Stato prossimo:

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

Se il livello di energia è alto (s_t = alto):
se scelgo Wait – s_{t+1} = alto.
se scelgo Search, s_{t+1} avrà una certa probabilità di diventare low.

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

Se il livello di energia è basso (s_t = basso):
se scelgo Wait – s_{t+1} = basso.
se scelgo Recharge - s_{t+1} = alto.
se scelgo Search, s_{t+1} avrà una certa probabilità di fermarsi.

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

Reward del Robot

Funzione Reward: $R_{s->s'|a} = E\{r_{t+1} = r' | s_t = s, a_t = a, s_{t+1} = s'\}$

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

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

-3 se occcorre portarlo a ricaricarsi.

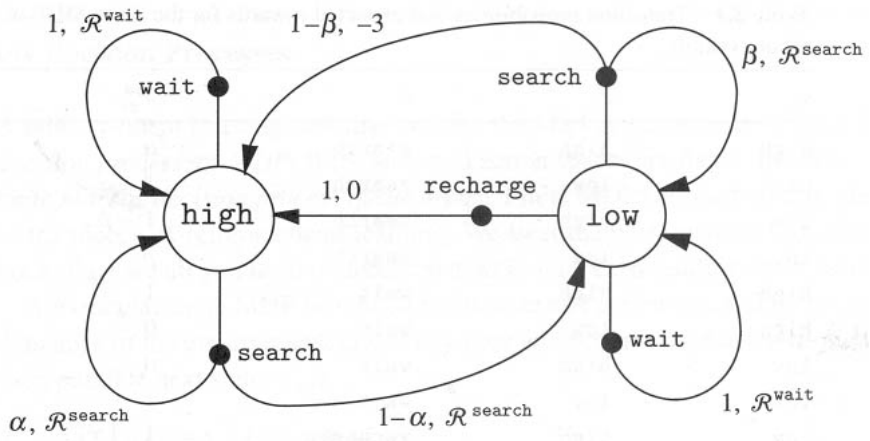
0 se il robot va autonomamente a ricaricarsi.

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

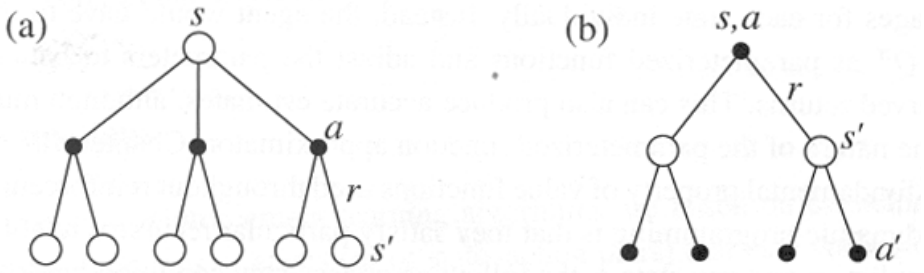
Forma tabellare

s	s'	a	$P_{s->s'|a}$	$R_{s->s'|a}$
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 ancora determinata.

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.

La value function

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

Vogliamo costruire agenti lungimiranti.

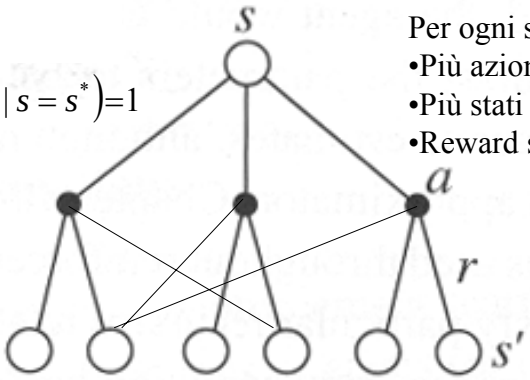
State-Value function

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

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

Value function e modelli markoviani

$$\sum_{j=1}^{N\_azioni} \Pr(a_j \mid s = s^*) = 1$$



Per ogni stato devo valutare:
- Più azioni.
- Più stati prossimi
- Reward stocastici.

$$\sum_{k=1}^{N\_stati} \Pr(s_{t+1} = s_k \mid s_t = s'; a_t = a_j) = 1$$

La Value function sulle azioni

La value function può riguardare le azioni.

Action-Value function

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

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

Dove abbiamo già trovato una Value function associata alle azioni?

Confronto con il setting non associativo

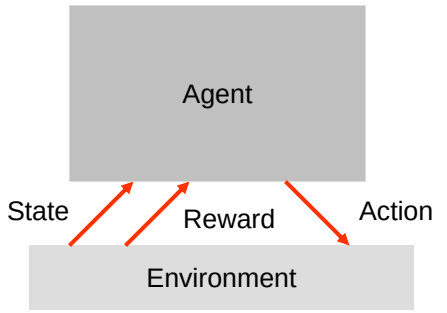
	Setting non associativo	Setting associativo
Task	Azioni	Comportamenti (catena di azioni)
Reward	Reward istantaneo	Somma (scontata) dei reward collezionati lungo il task.
Max	Reward atteso sulla singola azione	Reward del comportamento
Orizzonte temporale del task	Finito (1 azione)	Finito / infinito per il singolo task
Policy	Stocastica	Stocastica
Stato	Non definito	Markoviano

Reinforcement Learning Problem

Given: Repeatedly…
- Executed action
- Observed state
- Observed reward

Learn action policy $\pi: S \rightarrow A$
- Maximizes life reward
  $r_0 + \gamma r_1 + \gamma^2 r_2 \ldots$
  from any start state.
- Discount: $0 < \gamma < 1$

Note:
- Unsupervised learning
- Delayed reward



$$s_0 \xrightarrow[r_0]{a_0} s_1 \xrightarrow[r_1]{a_1} s_2 \xrightarrow[r_2]{a_2} s_3$$

Goal: Learn to choose actions that maximize life reward

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

How About Learning the Policy Directly?

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

Per ogni a*, valuto V(s)….

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.