

# 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  
[borgnese@di.unimi.it](mailto:borgnese@di.unimi.it)



A.A. 2013-2014

1/42



## Sommario



### Il Reinforcement Learning in setting non associativi

Processi Markoviani.

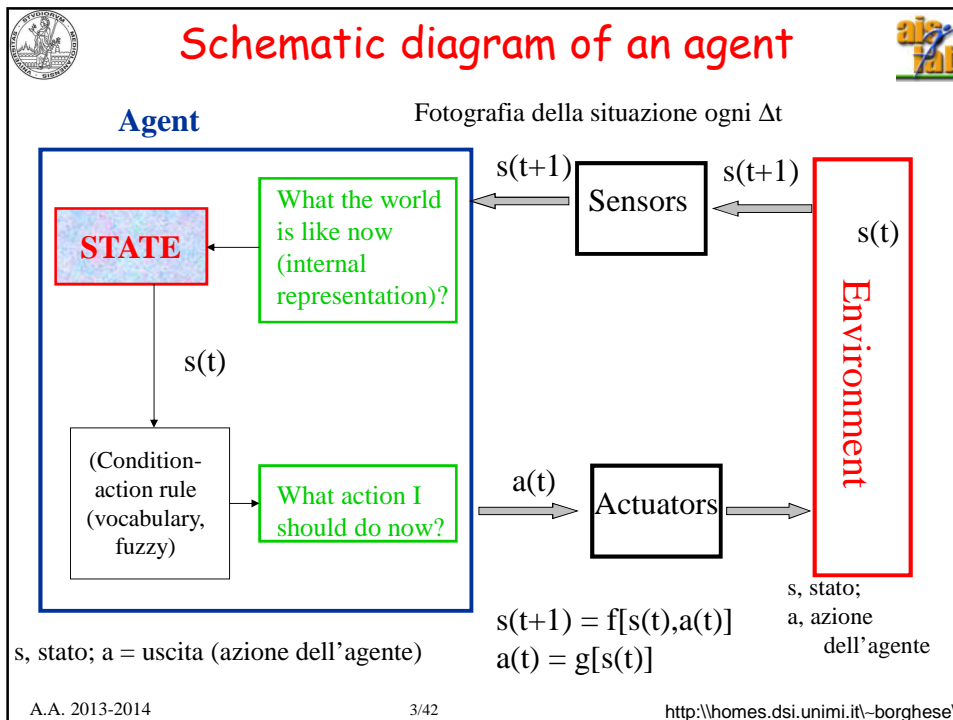
La value function: ricompensa a lungo termine: formulazione ricorsiva.

Esempi di calcolo

A.A. 2013-2014

2/42

<http://\borgnese.di.unimi.it>



**Il problema del "n-Armed bandit"**

Situazione iniziale costante.

Scelta tra n azioni (azione + reward esaurisce l'episodio)

La richiesta di scegliere viene ripetuta più volte nel tempo.

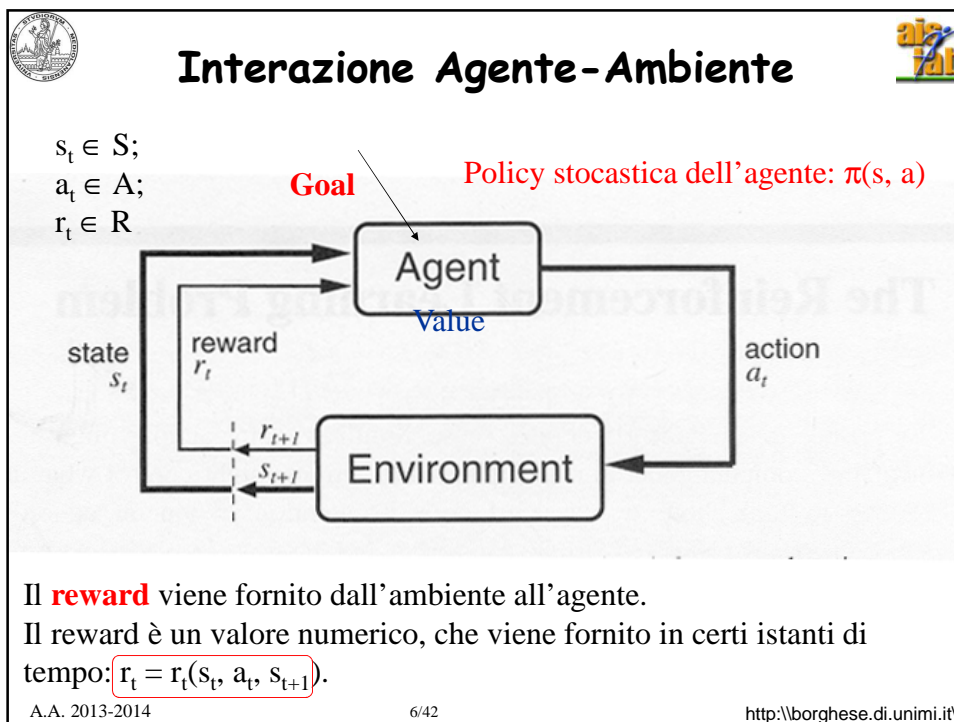
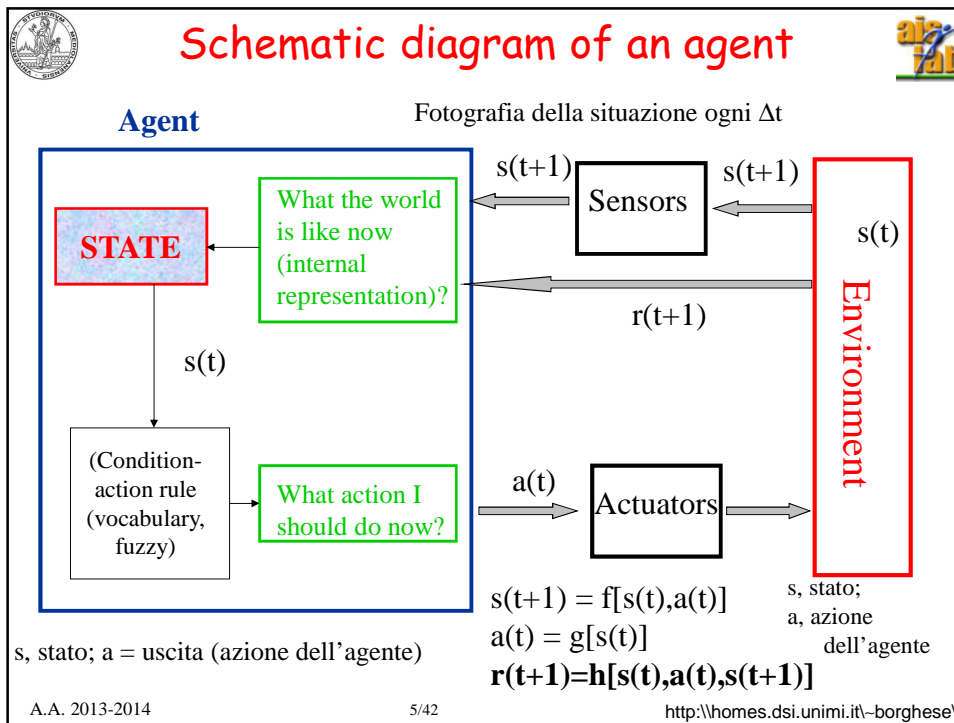
La ricompensa è stocastica (e.g. slot machine).

Obiettivo: viene massimizzata la ricompensa a lungo termine.

Soluzione possibile: selezionare l'azione che fornisce la massima ricompensa a lungo termine.

Come si può estendere la soluzione trovata ai comportamenti?

A.A. 2013-2014 4/42 <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.



## 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.

Policy = definisce quale azione in un certo stato (può essere stocastica, e.g.  $\epsilon$ -greedy)

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? Introduzione del concetto di **stato**.



## Reward e Obiettivi



Il reward è “esterno” all’agente.

Massimizzare la ricompensa a lungo termine, Value, cumulando le ricompense istantanee:  $r_t(a(t), s(t), s(t+1)) \in \mathbb{R}$ .

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:

$$R_t^\pi = r_{t+1} + r_{t+2} + r_{t+3} + \dots + \underbrace{r_T}_{\text{Terminal State}}$$

Quando è adeguata?

Problemi ad orizzonte finito (episodic tasks, a terminal state is defined).

Problemi stazionari.

Obiettivo:  $\pi' : R_t^{\pi'} \geq R_t^\pi$



## 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  $\gamma \rightarrow 1$ ?

A.A. 2013-2014

13/42

<http://borghese.di.unimi.it/>



## Reinforcement Learning Problem



Given: Repeatedly...

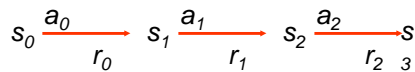
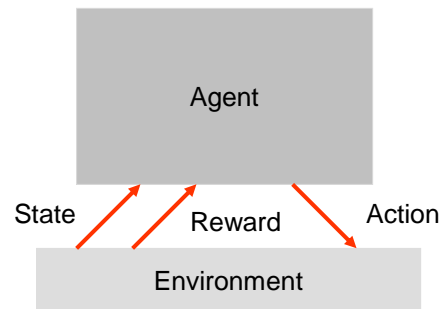
- Executed action
- Observed state
- Observed reward

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

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

A.A. 2013-2014

14/42

<http://borghese.di.unimi.it/>



## Il nostro filo logico

La Value function ci serve per decidere l'azione migliore.  
Per calcolare la Value function devo collezionare reward futuro.

Come se ne esce?

Determinazione algebrica della Value Function  
Determinazione "esplorativa" della Value Function  
Determinazione della Value Function e scelta della policy via via sempre  
più intrecciate tra loro.



## Sommario

Il Reinforcement Learning.

**Processi Markoviani.**

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

Esempi di calcolo





## 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.

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

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

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

A.A. 2013-2014

19/42

<http://borghese.di.unimi.it/>



## Approssimazione



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

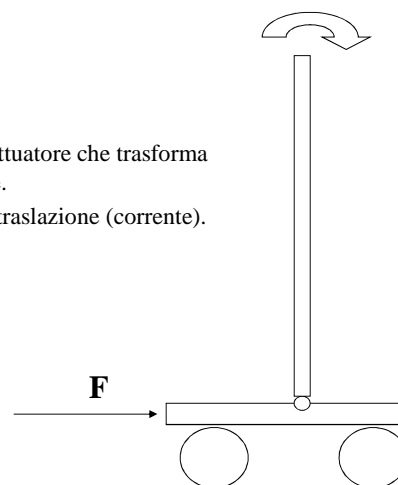
Environment – cart + pole.

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. 2013-2014

20/42

<http://borghese.di.unimi.it/>



## 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. 2013-2014

21/42

<http://borghese.di.unimi.it/>



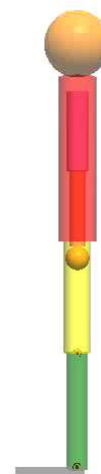
## Results on simulations



Hat model of the human body  
Muscles with maximum power  
and limited time constants.  
Control of the joints of the leg.

Reinforcement learning with  
reinforcement signal when  
falling.

Video: APPR\_tutto.m1v



A.A. 2013-2014

22/42



## Sommario



Il Reinforcement Learning.

Processi Markoviani.

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

Esempi di calcolo

A.A. 2013-2014

23/42

<http://\borghese.di.unimi.it/>



## 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:

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

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

**Goal:** collezionare il maggior numero di lattine.

A.A. 2013-2014

24/42

<http://\borghese.di.unimi.it/>



## 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. 2013-2014

25/42

<http://borghese.di.unimi.it/>



## Reward del Robot



### Funzione Reward:

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

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

$R^{\text{wait}}$  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. 2013-2014

26/42

<http://borghese.di.unimi.it/>



## Forma tabellare

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

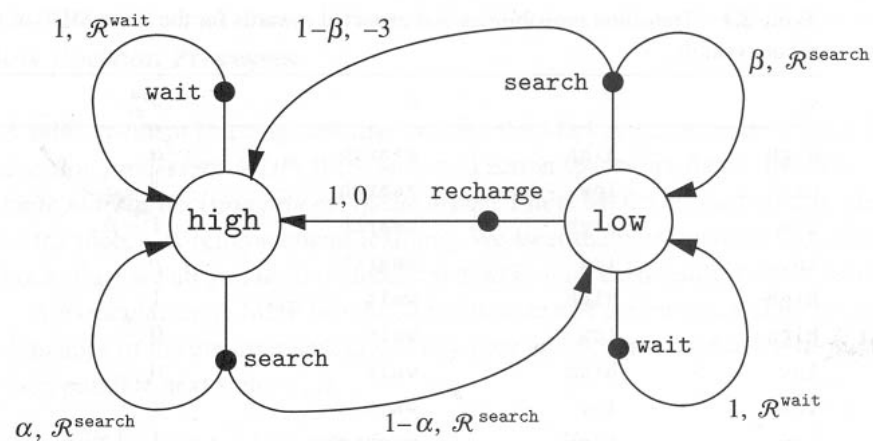
A.A. 2013-2014

27/42

<http://borghese.di.unimi.it/>



## State Transition Graph



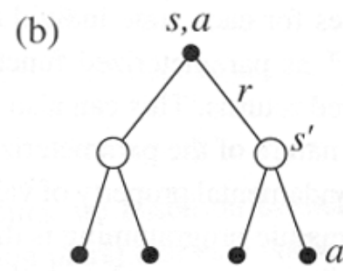
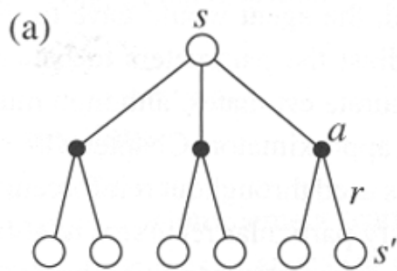
A.A. 2013-2014

28/42

<http://borghese.di.unimi.it/>



## L'albero delle decisioni



Guardo la sequenza  $s \rightarrow a \rightarrow s' \rightarrow a'$  in 1 passo di interazione

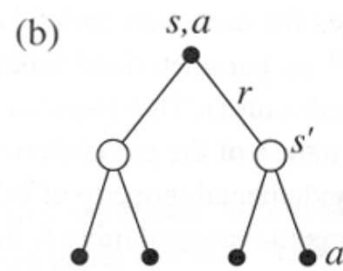
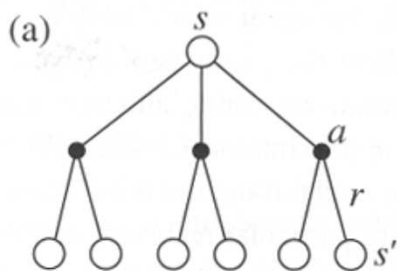
A.A. 2013-2014

29/42

<http://borghese.di.unimi.it/>



## 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.

A.A. 2013-2014

30/42

<http://borghese.di.unimi.it/>



## 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  |



## Il nostro filo logico



La Value function ci serve per decidere l'azione migliore.  
Per calcolare la Value function devo collezionare reward futuro.

Come se ne esce?

Determinazione algebrica della Value Function  
Determinazione "esplorativa" della Value Function  
Determinazione della Value Function e scelta della policy via via sempre più intrecciate tra loro.





## Sommario



Il Reinforcement Learning.

Processi Markoviani.

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

Esempi di calcolo

A.A. 2013-2014

33/42

<http://borghese.di.unimi.it/>



## Value function & policy



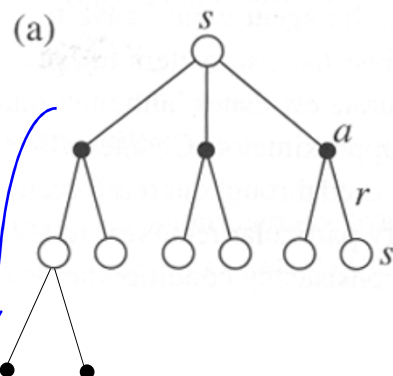
Nulla è detto sulla policy: dato uno stato, in quale nodo azione mi sposto?

Vogliamo costruire agenti lungimiranti.

State-Value function  
(function of the policy):

$$Q^\pi(s, a) = E_\pi \{ R_t \mid s_t = s; a_t = a \} =$$

$$E_\pi \left\{ \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid s_t = s; a_t = a \right\}$$



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

A.A. 2013-2014

34/42

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



## Value function e modelli markoviani

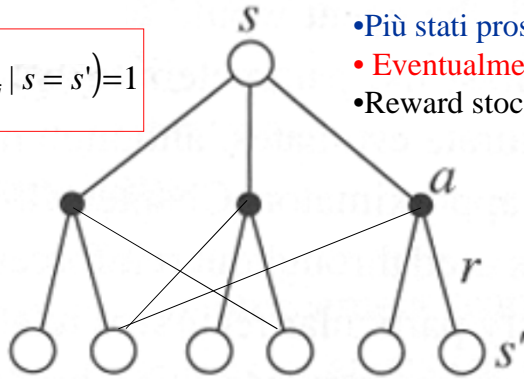


Anche la policy può essere stocastica. Per ogni coppia stato-azione

devo valutare:

- Più stati prossimi
- Eventualmente più azioni in  $s'$
- Reward stocastici.

$$\sum_{j=1}^{N_{\text{azioni}}} \Pr(a'_j | s = s') = 1$$



$$\sum_{k=1}^{N_{\text{stati}}} \Pr(s_{t+1} = s_k | s_t = s'; a_t = a_j) = 1$$

A.A. 2013-2014

35/42

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



## Esempio di calcolo della Value function



Value function

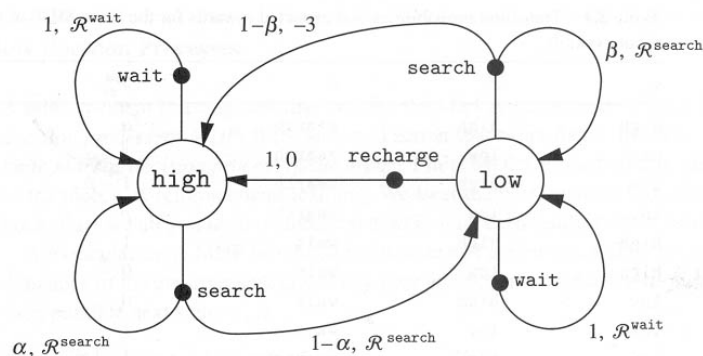
$Q(\text{high}, \text{wait}) = ?$

$Q(\text{low}, \text{search}) = ?$

Policy

$a(\text{high}) = \text{wait}$


$a(\text{low}) = \text{search}$




A.A. 2013-2014


36/42

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



## Policy deterministica - I





$\alpha=0.4, \beta=0.1, \gamma=0.8,$   
 $R_{search}=3, R_{wait}=1, R_{recharge} = -3, R_{auto} = 0$   
 $s = \text{High} - a = \text{Wait};$   
 $s = \text{Low} - a = \text{Search};$

$Q(h,w) = [1+0.8 Q(h,w)]$   
*Wait*  
 $Q(l,s) = 0.1x[3+0.8xQ(l,s)]+0.9x[-3+0.8Q(h,w)]$

↖


Search -> Low

↗


Search -> High


**Sistema lineare di 2 equazioni nelle 2 incognite: Q(h,w) e Q(l,s)**  
**Come calcolo Q(h,w) e Q(l,s)?**

A.A. 2013-2014
37/42
<http://homes.dsi.unimi.it/~borghese/>



## Policy deterministica - I





$\alpha=0.4, \beta=0.1, \gamma=0.8,$   
 $R_{search}=3, R_{wait}=1, R_{recharge} = -3, R_{auto} = 0$   
 $s = \text{High} - a = \text{Wait};$   
 $s = \text{Low} - a = \text{Search};$

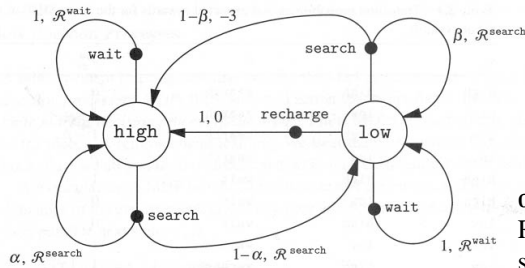
$Q(h,w) = [1+0.8 Q(h,w)] \quad \rightarrow Q(h,w) = 5$   
 $Q(l,s) = 0.1x[3+0.8xQ(l,s)]+0.9x[-3+0.8Q(h,w)]$

$Q(l,s) = 0.3 + 0.08 Q(l,s) - 2.7 + 0.72Q(h,w) \rightarrow$   
 $Q(l,s) (1-0.08) = 0.3-2.7+3.6 \quad \rightarrow \quad Q(l,s) = 1,304$

A.A. 2013-2014
38/42
<http://homes.dsi.unimi.it/~borghese/>



## Policy deterministica - II



$\alpha=0.4, \beta=0.1, \gamma=0.8,$   
 $R^{search}=3, R^{wait}=1, R^{recharge} = -3, R^{auto} = 0$   
 s = High - a = Search;  
 s = Low - a = Search;

$$Q(h,s) = 0.4x [3+0.8 Q(h,s)] + 0.6 x [3+0.8 x [Q(l,s)]]$$

$$Q(l,s) = 0.1x [3+0.8xQ(l,s)] + 0.9x [-3+0.8Q(h,s)]$$

**Sistema lineare di 2 equazioni nelle 2 incognite:  $Q(h,s)$  e  $Q(l,s)$**

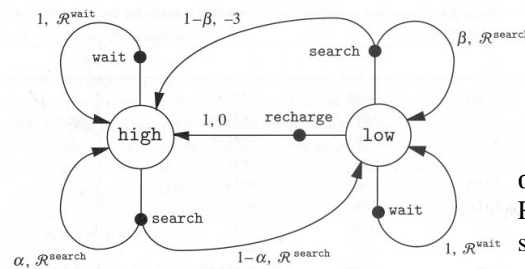
A.A. 2013-2014

39/42

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



## Policy deterministica - II



$\alpha=0.4, \beta=0.1, \gamma=0.8,$   
 $R^{search}=3, R^{wait}=1, R^{recharge} = -3, R^{auto} = 0$   
 s = High - a = Wait;  
 s = Low - a = Search;

$$Q(h,s)[1-0.32] = 1.2 + 1.8 + 0.48 x Q(l,s)$$

$$Q(l,s)[1-0.08] = 0.3-2.7+ 0.72Q(h,s)$$

$$Q(h,s)[0.68] - 0.48 Q(l,s) = 3.$$

$$\rightarrow Q(h,s) = 5.7429$$

$$Q(l,s)[0.92] - 0.72Q(h,s) = -2.4$$

$$\rightarrow Q(l,s) = 1.8857$$

La policy è migliore per entrambi gli stati.

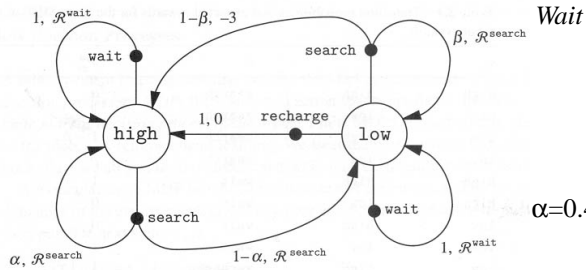
A.A. 2013-2014

40/42

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



## Policy stocastica



$\alpha=0.4, \beta=0.1, \gamma=0.8, \mathcal{R}^{\text{search}}=3, \mathcal{R}^{\text{wait}}=1$

$$Q(h,w) = 1x\{1+0.8 [\Pr(\text{wait} | h) Q(h,w) + \Pr(\text{search} | h) Q(h,s)]$$

$$Q(h,s) = 0.1x\{3+0.8 x [\Pr(\text{search} | l) Q(l,s) + \Pr(\text{wait} | l) Q(l,w) + \Pr(\text{recharge} | l)Q(l,r)] \} + 0.9 x \{-3+ 0.8 [\Pr(\text{wait} | h) Q(h,w) + \Pr(\text{search} | h) Q(h,s)]\}$$

.....

**Sistema lineare di 5 equazioni nelle 5 incognite:**

**$Q(h,w), Q(h,s), Q(l,w), Q(l,s), Q(l,r)$  Come calcolo  $V_h$  e  $V_l$ ?**

A.A. 2013-2014

41/42

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



## Sommario



Il Reinforcement Learning.

Processi Markoviani.

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

Esempi di calcolo

A.A. 2013-2014

42/42

<http://borghese.di.unimi.it/>