

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 Informatica

Alberto.borghese@unimi.it

Chapter 3 – Barto Sutton



Sommario



Agenti e sistemi dinamici

La value function: ricompensa a lungo termine.

Esempi di calcolo



Gli agenti



Agente (software): essere software che svolge servizi per conto di un altro programma, solitamente in modo automatico ed invisibile. Tali software vengono anche detti agenti intelligenti

“They are seen as a *natural metaphor* for conceptualising and building a wide range of complex computer systems (the world contains many passive objects, but it also contains very many *active* components as well);

They *cut across a wide range of different technology and application areas*, including telecoms, human-computer interfaces, distributed systems, WEB and so on;

They are seen as a natural development in the search for ever-more powerful abstractions with which to build computer systems.“

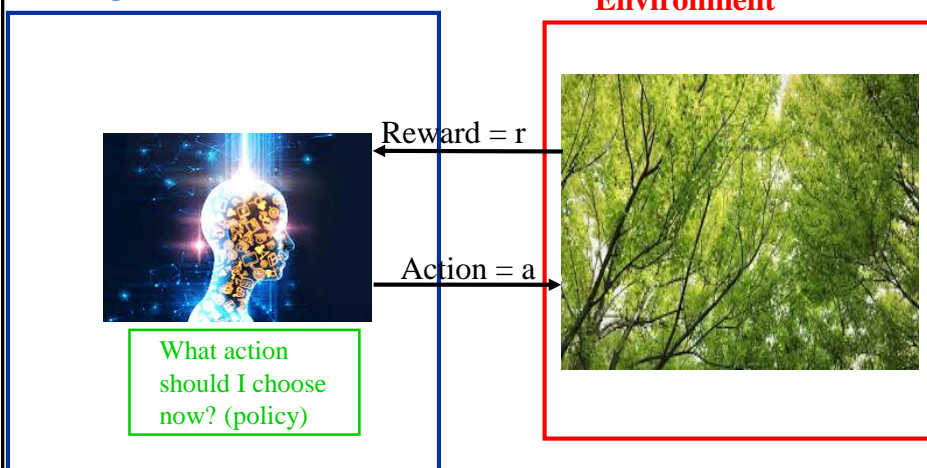


Schematic diagram of an agent



Agent

Environment



Se a dipende dalla situazione?



The same action, different outcomes



white wins



white loses

It depends on **the situation** (configuration of the pieces on the board)



Evoluzione del sistema



Agent

Environment



What action should I choose now? (policy)

Reward = r_{t+1}

Stato = s_{t+1}

Action $a_t = g(s_t)$



$s_t \rightarrow s_{t+1}$
 $s_{t+1} = f(s_t, a_t)$

L'azione dipende dalla **situazione**: $a_t = g(s_t)$!

La situazione dell'ambiente evolve nel tempo: **sistema dinamico**

$s_{t+1} = f(s_t, a_t)$



Lo stato

Lo stato è la situazione in cui si trova l'ambiente.

La situazione è il risultato dell'azione dell'agente sull'ambiente e dalla dinamica (**spesso non nota**) dell'ambiente stesso.

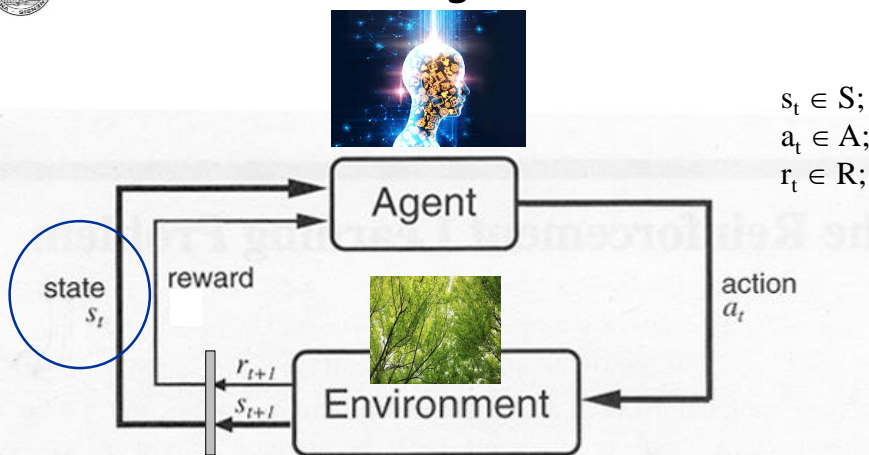
- La situazione è quindi il risultato della storia passata dell'interazione con l'ambiente.
- Lo stato deve essere:
 - efficiente per il raggiungimento del goal.
 - misurato dall'agente (osservabile).

Come rappresento lo stato, s ?

- Memorizzo la sequenza temporale degli stimoli semplici di interesse. Utilizzo una variabile che riassume la situazione attuale (concetto di **stato**).

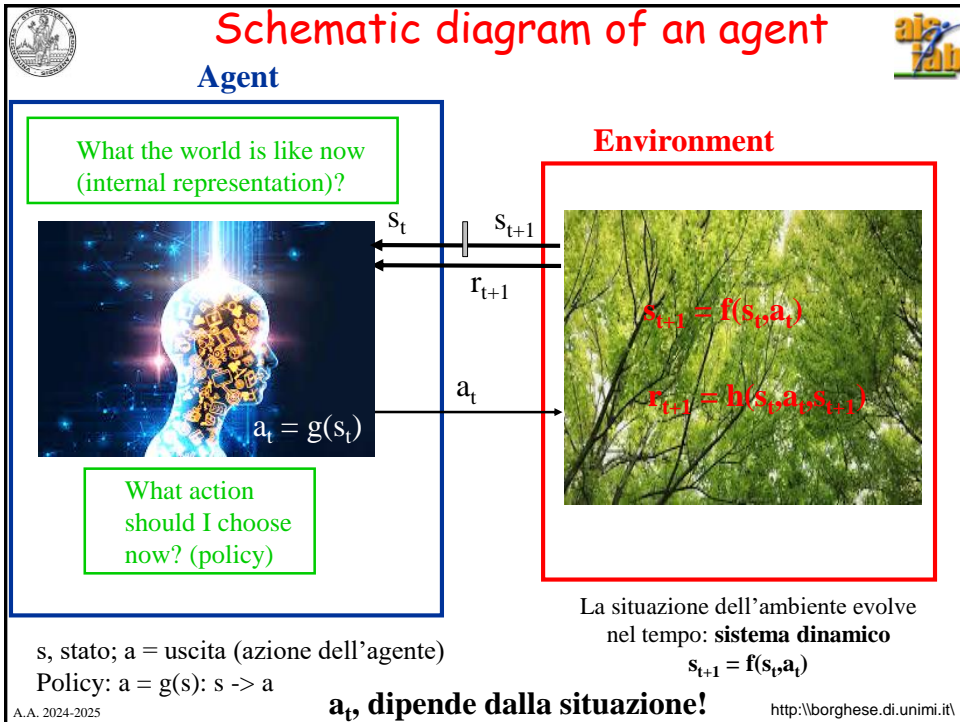


Interazione Agente-Ambiente



Il **reward** viene fornito dall'ambiente all'agente.

Il reward è un valore numerico, qualitativo, che viene fornito in certi istanti di tempo: $r_{t+1} = r(s_t, a_t, s_{t+1})$.



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. Si definisce **processo stocastico markoviano**, un processo aleatorio in cui la **probabilità di transizione** che determina il passaggio a uno stato di sistema **dipende solo dallo stato del sistema, immediatamente precedente** e non da come si è giunti a questo stato.

Andrej Markov (1856-1922)

Formalizziamo. Supponiamo s_t e r_t variabili discrete appartenenti a un insieme finito di valori.

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

Se lo stato è Markoviano:
 $Pr\{s_{t+1} = s' | s_t, a_t\}$

NB: Nella pratica $Pr\{s_{t+1} = s' | s_t, a_t\}$ non è nota!

Agent

What the world is like now (internal representation)?

$a_t = g(s_t)$

What action should I choose now? (policy)

Environment

$s_{t+1} = f(s_t, a_t)$

$r_{t+1} = b(s_t, a_t, s_{t+1})$

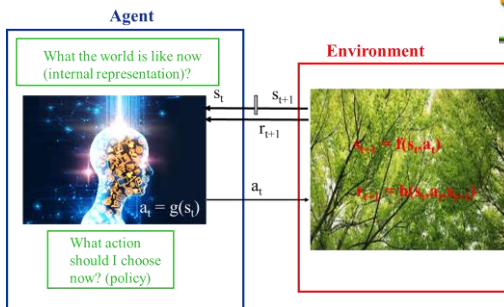
A.A. 2024-2025 10/55 <http://borghese.di.unimi.it/>



Rinforzo 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:

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

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

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

Elemento chiave è lo stato (situazione)

Tutto quanto precede lo stato s_t non influenza la statistica.



Sommario



Agenti e sistemi dinamici

La value function: ricompensa a lungo termine

Esempi di calcolo



Il condizionamento classico



Il segnale di rinforzo è sempre lo stesso per ogni coppia input – output: **Reward istantaneo, non dipende dalla situazione.**

Esempio: risposta riflessa Pavloviana. Campanello (stimolo condizionante) prelude al cibo. Questo induce una risposta (salivazione). La risposta riflessa a uno stimolo viene evocata da uno stimolo condizionante.





Il rinforzo



“Learning is an adaptive change of behaviour and what is indeed the reason of its existence in animals and man” (K. Lorenz, 1977).

Rinforzo **puntuale istante per istante**, associato all'azione (**condizionamento classico**).

Rinforzo **puntuale istante per istante**, associato a un'azione in una situazione. La situazione è definita da **catena di azioni (comportamento)** - (**condizionamento operante**).

Rinforzo non necessariamente a ogni istante),

Il rinforzo deve essere associato anche alla situazione (stato).

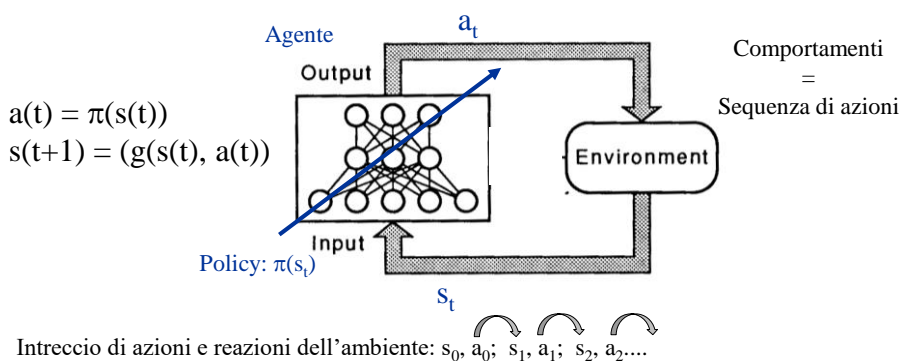



Comportamento




Interessa un **comportamento**. Una **sequenza di input / output** che può essere modificata agendo sui parametri che definiscono il **comportamento dell'agente**: l'azione scelta in un certo stato (= situazione). Il comportamento dell'agente viene chiamato **policy**, $\pi(s)$.

L'**agente** deve scoprire quale azione (**policy**) fornisca la ricompensa massima provando le varie azioni (in stile trial-and-error) sull'**ambiente** con un criterio intelligente.






Apprendimento



L'agente vuole scegliere a in modo tale da massimizzare il **reward TOTALE** accumulato **nel tempo**. A partire dallo stato (situazione) in cui si trova deve scegliere il comportamento (le azioni) che massimizzano il **reward a lungo termine** a partire dal **reward istantaneo**, r_{t+1} .

Agent


What the world is like now
(internal representation)?



What action
should I choose
now? (policy)


$a_t = g(s_t)$

Environment




Vogliamo costruire agenti lungimiranti che non siano greedy e miopi

A.A. 2024-2025
17/55
<http://borghese.di.unimi.it/>



Ruolo dell'agente



Può scegliere un'azione sull'ambiente, tra un insieme continuo o discrete di azioni (**policy**)


L'azione dipende dalla situazione. La situazione è riassunta dallo **stato** del sistema.

La scelta dell'azione è non banale non può scegliere semplicemente l'azione che massimizza il reward istantaneo, richiede un certo grado di "intelligenza".

L'agente ha un'intelligenza legata alla memoria. Non può mantenere in memoria tutto il passato.

Agent


What the world is like now
(internal representation)?



What action
should I choose
now? (policy)

$a_t = g(s_t)$

Environment



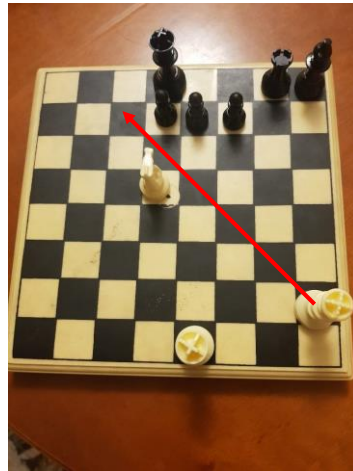
A.A. 2024-2025
18



Esempio



white gains a knight at time t ($r > 0$)
white loses (bad total reward)



white gains nothing at time t ($r = 0$)
white wins (good total reward)

Deve guardare oltre la mossa greedy

A.A. 2024-2025

19/55

<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. 2024-2025

20/55

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



Results on simulations

HAT (Head + Arms + Trunk) model of the human body

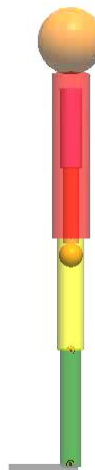
Muscles with maximum power and limited time constants.

Control of the joints of the leg (ankle, knee and hip).

Reinforcement learning with reinforcement signal ONLY when falling down.

Hip-ankle vs Knee control

Video: APPR_tutto.m v



Value function

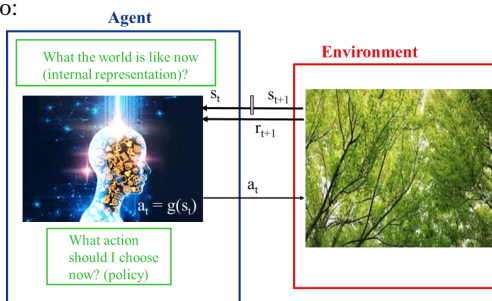
Cosa si intende per ricompensa a lungo termine?

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:

$$R_t^\pi = r_{t+1} + r_{t+2} + r_{t+3} + r_{t+4} + \dots$$

$$R_t^\pi = \sum_{k=0}^{\infty} r_{t+k+1}$$



L'agente vuole massimizzare il **valore atteso del reward totale** (somma dei reward a partire dallo stato in cui si trova). **Come determino i reward futuri?**



Infinite horizon problems (continuing tasks)



- We introduce a **discounted reward**: we give more weight to the rewards closer in the future and avoid that total reward diverges.

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”.

R_t is the **value at time t**, of rewards that will be **collected in the future**.

$$R_t \leq \frac{z}{1 - \gamma} \quad \text{if } r_{t+k} \leq z \quad \forall k$$

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

Simile nel considerare via via sempre meno i reward lontani dal tempo t corrente.

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



Problemi a orizzonte temporale finito



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:

$$R_{N+1} = r_{N+1} + \gamma r_{N+2} + \gamma^2 r_{N+3} + \dots + \gamma^{t-1} r_T$$

Terminal State

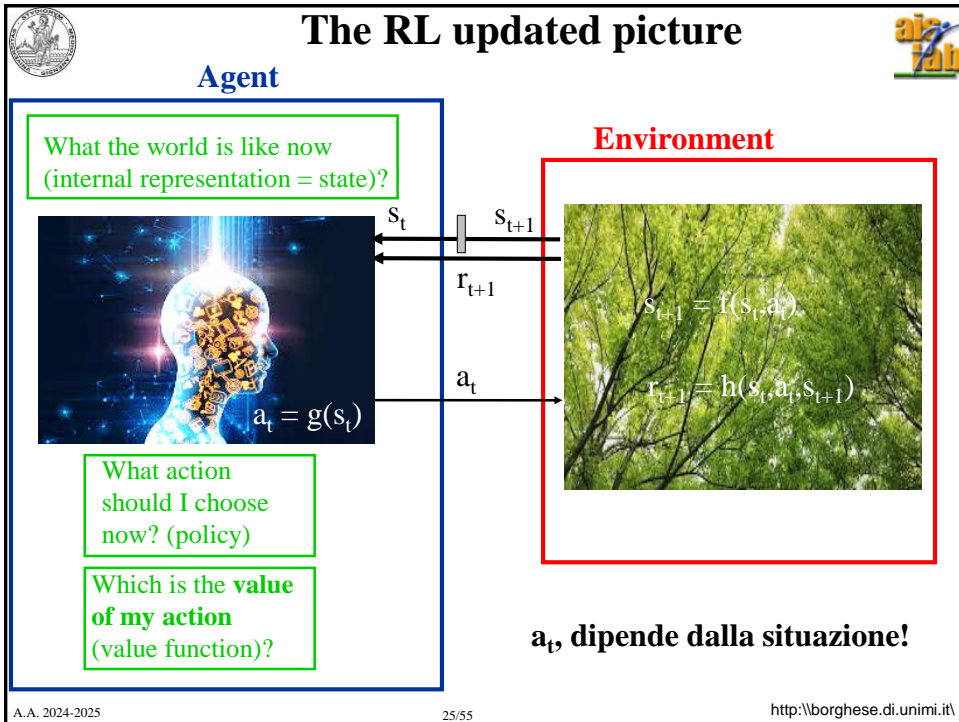
Quando è adeguata?

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



Problemi stazionari.

Non ci sarebbe bisogno del discount, γ .

Obiettivo migliorare la policy: π' : $R^{\pi'} > R^\pi$



Gli elementi del RL

Environment. Fornisce il reward (istantaneo), fornisce lo stato dell'ambiente. Reagisce alle azioni (output) dell'agente.

Agente. Ragiona sullo stato fornito dall'ambiente e sul reward istantaneo per produrre un'azione adeguata sull'ambiente.

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. 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. Ricompensa **lungo termine**. Somma dei reward: costi associati alle azioni scelte istante per istante + costo associato allo stato finale. **Orizzonte temporale ampio**. Rinforzo secondario. Vogliamo realizzare agenti che **ragionino in modo strategico**. (cf. "Cost-to-go" in ricerca operativa). Valore atteso della somma dei reward associati al comportamento.

A.A. 2024-2025

26/55

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



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. ϵ -greedy)

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

Reward (rinforzo primario) = viene generato all'esterno dell'agente.

Value function (rinforzo secondario) = viene stimata all'interno dell'agente.



Reward e Obiettivi



Il reward è "esterno" all'agente.

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

Viene definita una ricompensa che viene massimizzata.

Per raggiungere un goal, la ricompensa del goal deve essere sufficientemente grande.

L'agente impara il task (raggiunge il goal) massimizzando la ricompensa.

Collegamento tra reward e goal.

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



Proprietà del rinforzo



L'ambiente o l'interazione può essere complessa.

Il rinforzo può avvenire dopo una sequenza più o meno lunga di interazioni (azione-> cambiamento di stato e reward istantaneo): **delayed reward**.

E.g. Agente -- giocatore di scacchi
 Ambiente – avversario

Problemi collegati:

temporal credit assignement (quando ho sbagliato la mossa?)

structural credit assignement (quale mossa era sbagliata?)

L'apprendimento non è per esempi, ma dall'osservazione dell'impatto sull'ambiente del comportamento dell'agente.



Meccanismo di apprendimento nel RL



Inizializzazione: se l'agente non agisce sull'ambiente non succede nulla. Occorre specificare una policy iniziale.

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

- 1) Implemento una policy ($\pi(s,a)$)
- 2) Stimo la Value function ($Q^\pi(s,a)$) per tutte le coppie stato-azione
- 3) Miglioro la policy, $\pi(s,a)$.

Quando miglio la policy, devo stimare la nuova Value Function, è cambiata la policy.



Meccanismo di apprendimento nel RL



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

- 1) Implemento una policy ($\pi(s,a)$)
- 2) Stimo la Value function ($Q^\pi(s,a)$) per tutte le coppie stato-azione
- 3) Miglioro la policy, $\pi(s,a)$.

Framework analizzati per calcolare $Q^\pi(s,a)$:

1. Stocastico completo. **L'agente conosce la statistica della dinamica dell'ambiente e dei reward**, scrive **le equazioni lineari** che legano le value function in stati diversi e **calcola** i valori di $Q^\pi(\cdot)$.
2. Stocastico con aggiornamento. **L'agente conosce la statistica della dinamica dell'ambiente e dei reward**. **Procede da uno stato iniziale a quello finale**. Da ogni stato **esplora in parallelo tutti i possibili stati prossimi** e aggiorna i valori delle $Q^\pi(\cdot)$.
3. Stocastico con interazione singola e con la scelta di una singola azione. **L'agente NON conosce la statistica della dinamica dell'ambiente e dei reward**. **Procede da uno stato iniziale a quello finale**. Da ogni stato esplora una sola azione e **un SOLO solo stato prossimo**. Aggiorna i valori di $Q^\pi(\cdot)$.



Sommario



Il Reinforcement Learning.

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

Esempi di calcolo



Framework stocastico completo



Inizializzazione: se l'agente non agisce sull'ambiente non succede nulla. Occorre specificare una policy iniziale.

Ciclo: stima di $Q^\pi(\cdot)$ e aggiornamento di $\pi(s,a)$.

L'agente conosce la statistica della dinamica dell'ambiente e dei reward, scrive **le equazioni lineari** che legano le value function in stati diversi e **calcola** i valori di $Q^\pi(\cdot)$.

Migliora la policy, $\pi(s,a)$.



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.

Azioni ammissibili (policy):

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

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

Goal: collezionare il maggior numero di lattine.



Funzionamento del Robot - ($s_t = \text{alto}$)



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

Funzione Stato prossimo se il livello di energia è alto, $s_t = \text{alto}$. 2 possibili azioni:

1) se scelgo $a = \text{Wait}$ $\rightarrow s_{t+1} = \text{alto}$.

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

2) se scelgo $a = \text{Search}$, s_{t+1} avrà una certa probabilità α di rimanere nello stato high.

$$P_{\text{high} \rightarrow \text{high} | \text{search}} = Pr\{s_{t+1} = \text{high} | s_t = \text{high}, a_t = \text{search}\} = \alpha = 0.4$$

$$P_{\text{high} \rightarrow \text{low} | \text{search}} = Pr\{s_{t+1} = \text{low} | s_t = \text{high}, a_t = \text{search}\} = 1 - \alpha = 0.6$$



Funzionamento del Robot - ($s_t = \text{basso}$)



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

Funzione Stato prossimo se il livello di energia è basso ($s_t = \text{basso}$). 3 possibili azioni:

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

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

2) se scelgo $a = \text{Search}$, s_{t+1} avrà una certa probabilità β di rimanere low.

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

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

Il robot "muore" e viene portato a ricaricarsi.

3) se scelgo $a = \text{Recharge}$, s_{t+1} sarà carico: high.

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



Reward del Robot



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

Funzione Reward:

R^{search} reward se il robot sta cercando

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

-3 se occorre portarlo a ricaricarsi.

0 se il robot va autonomamente a ricaricarsi.

$g(\text{low}) = \text{search} \ \& \ s_{t+1} = \text{low}; \ g(\text{high}) = \text{search};$

$g(\text{low}) = \text{wait}; \ g(\text{high}) = \text{wait};$

$g(\text{low}) = \text{search}; \ \text{and} \ s_{t+1} = \text{high};$

$g(\text{low}) = \text{recharge};$

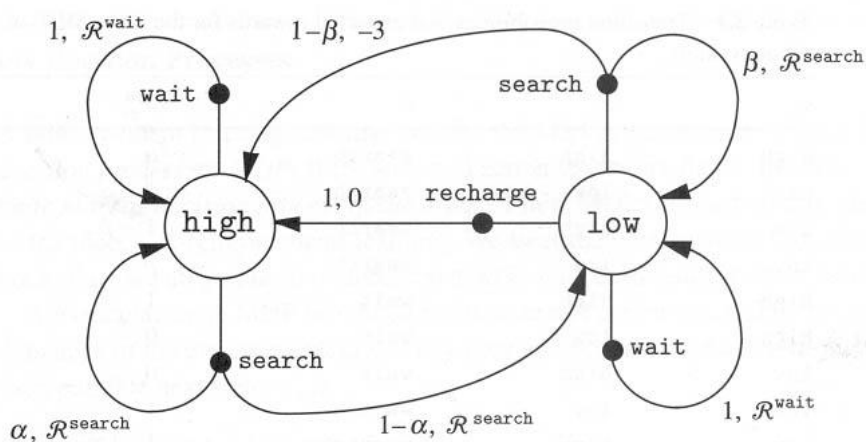
A.A. 2024-2025

37/55

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



State Transition Graph



Graphical model

Per ottenere il reward a lungo termine deve visitare il grafo (che ha un numero di stati finito)

A.A. 2024-2025

38/55

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



State Transition Table

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

A.A. 2024-2025

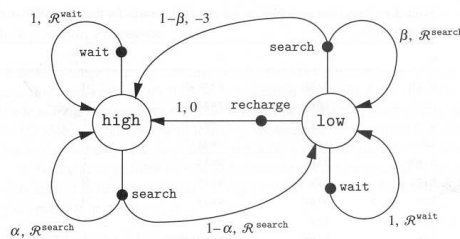
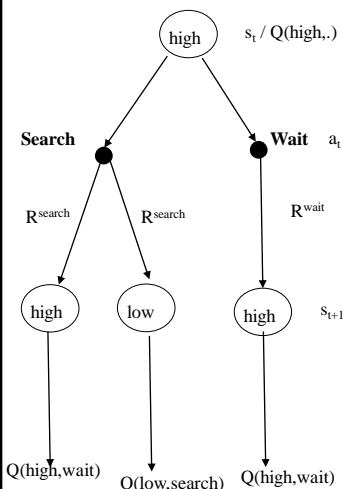
39/55

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

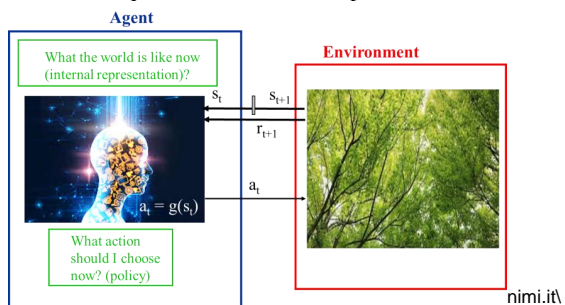


L'albero delle decisioni

Apri il grafo





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



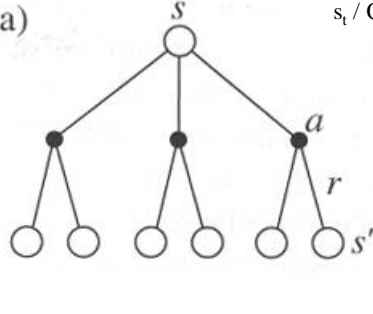
A.A. 2024-2025

nimi.it/

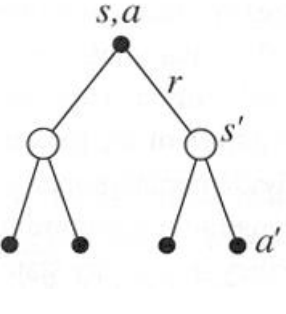
Policy

(a)

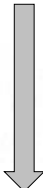


$s_t / Q^\pi(s,a)$

(b)



Max R_t





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. 2024-2025
41/55
<http://borghese.di.unimi.it/>

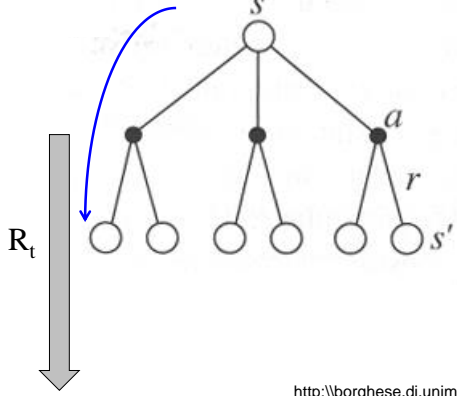
Value function & policy

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

Vogliamo costruire agenti lungimiranti.

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

Value function on state-action



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

A.A. 2024-2025
42/55
<http://borghese.di.unimi.it/>



Value function e modelli markoviani

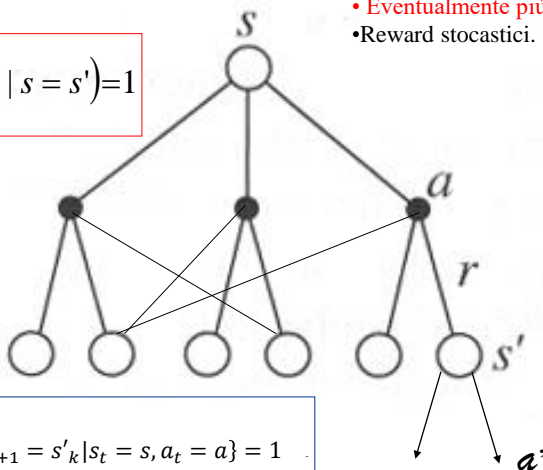


Anche la policy può essere stocastica.

Per ogni coppia stato-azione devo valutare:

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

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



$$\sum_{k=1}^{N_{\text{stati}}} \Pr\{s_{t+1} = s'_k | s_t = s, a_t = a\} = 1$$

A.A. 2024-2025

43/55

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



Il nostro filo logico



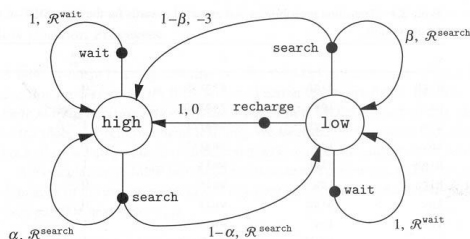
La Value function ci serve per decidere l'azione migliore.

Per calcolare la Value function devo collezionare reward futuro.

Per muovermi nel futuro devo scegliere delle azioni -> una policy

Come se ne esce?

Doteremo l'agente di un'intelligenza adatta a effettuare l'esplorazione.



A.A. 2024-2025

44/55

<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	Reward istantaneo (= 0 per certi passi di iterazione)
Max	Reward atteso sulla singola azione	Reward atteso sulla singola azione, dato un certo comportamento. Somma (scontata) dei reward collezionati lungo il task.
Orizzonte temporale del task	Finito (1 azione)	Finito (N azioni) / Infinito per il singolo task
Policy	Deterministica / Stocastica	Deterministica / Stocastica
Stato	Non definito	Markoviano

A.A. 2024-2025

45/55

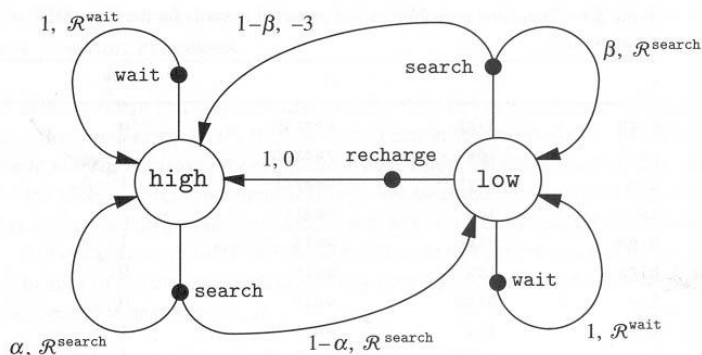
<http://borghese.di.unimi.it/>



Esempio di calcolo della Value function in $s = \text{high}$




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



A.A. 2024-2025

46/55

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



Agent waits in high state

Search

```

graph TD
    S((high)) -- Search --> L((low))
    S -- Search --> H((high))
    L -- Search --> L
    H -- Search --> H
    S -- Wait --> H2((high))
    H2 -- Wait --> H2
    
```

$s_t / Q(\text{high}, \text{wait})$

a_t

R^{search}

R^{wait}

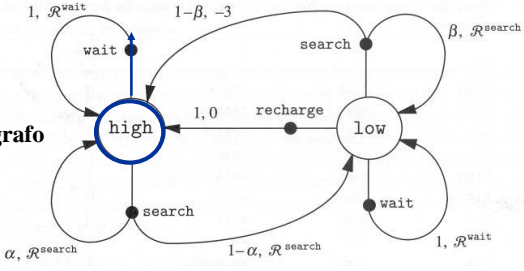
s_{t+1}

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

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

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


Apro il grafo



$\alpha=0.4, \beta=0.1, \gamma=0.8,$
 $R^{\text{search}}=3, R^{\text{wait}}=1, R^{\text{dead}}=-3, R^{\text{auto}}=0$

$Q(\text{high}, \text{wait}) = [R^{\text{wait}} + \gamma Q(\text{high}, \text{wait})] =$
 $[1 + 0.8 Q(\text{high}, \text{wait})] = 5$

A.A. 2024-2025 47/55 <http://borghese.di.unimi.it/>



Agent searches in high state

Search

```

graph TD
    S((high)) -- Search --> L((low))
    S -- Search --> H((high))
    L -- Search --> L
    H -- Search --> H
    S -- Wait --> H2((high))
    H2 -- Wait --> H2
    
```

$s_t / Q(\text{high}, \text{search})$

a_t

R^{search}

R^{wait}

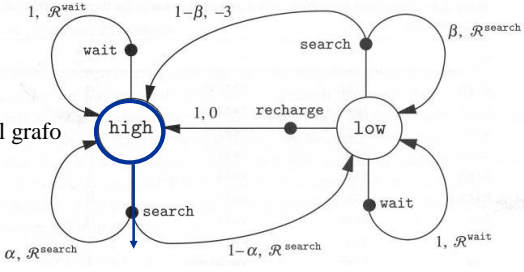
s_{t+1}

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

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

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

Apro il grafo



$\alpha=0.4, \beta=0.1, \gamma=0.8,$
 $R^{\text{search}}=3, R^{\text{wait}}=1, R^{\text{dead}}=-3, R^{\text{auto}}=0$

$Q(\text{high}, \text{search}) = \alpha [R^{\text{search}} + \gamma Q(\text{high}, \text{search})] +$
 $(1-\alpha) [R^{\text{search}} + \gamma Q(\text{low}, \text{search})] =$

$Q(\text{high}, \text{search}) = 0.4 [3 + 0.8 Q(\text{high}, \text{search})] +$
 $0.6 [3 + 0.8 Q(\text{low}, \text{search})]$

A.A. 2024-2025 48/55 <http://borghese.di.unimi.it/>

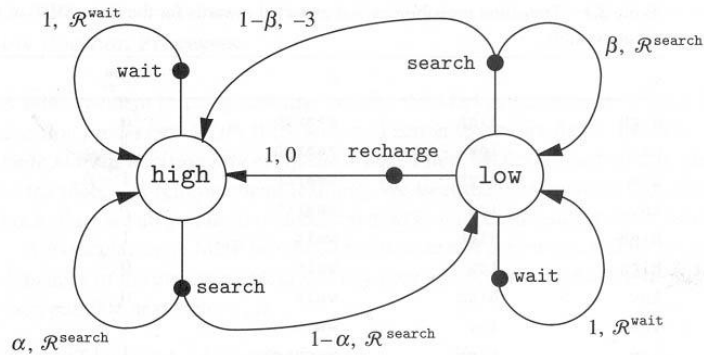


Esempio di calcolo della Value function



Policy deterministica
 $a(\text{high}) = \text{wait}$
 $a(\text{low}) = \text{search}$

Value function
 $Q(\text{high}, \text{wait}) = ?$
 $Q(\text{low}, \text{search}) = ?$



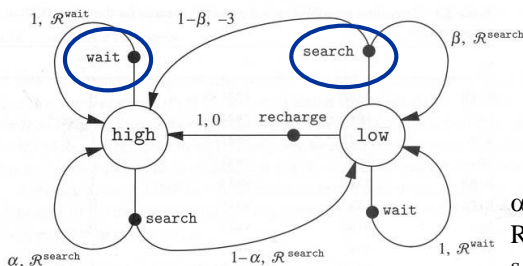
A.A. 2024-2025

49/55

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



Policy deterministica 1



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

$$Q(h,w) = [R_{\text{wait}} + \gamma Q(h,w)] \quad \text{Wait}$$

$$Q(l,s) = \beta x [R_{\text{search}} + \gamma Q(l,s)] + (1-\beta) x [R_{\text{dead}} + \gamma Q(h,w)] = 0.1 [3 + 0.8Q(l,s)] + 0.9 [-3 + 0.8Q(h,w)]$$

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 $Q(l,s)$?

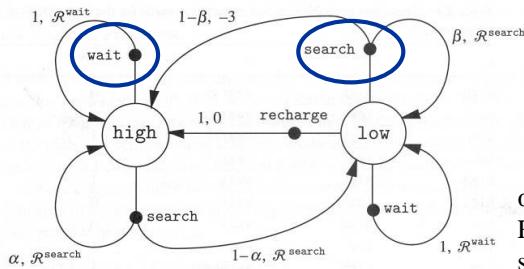
A.A. 2024-2025

50/55

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



Policy deterministica 1



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

$$Q(h,w) = [1+0.8 Q(h,w)] \quad \rightarrow Q(h,w) = 5$$

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

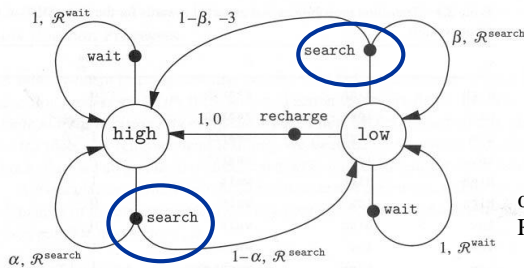
↓

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

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



Policy deterministica 2



$\alpha=0.4, \beta=0.1, \gamma=0.8,$
 $R^{\text{search}}=3, R^{\text{wait}}=1, R^{\text{dead}} = -3, R^{\text{auto}} = 0$

$s = \text{High} \rightarrow a = \text{Search};$
 $s = \text{Low} \rightarrow a = \text{Search};$

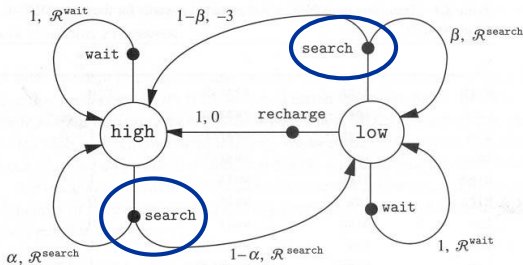
$$Q(h,s) = 0.4x [3+0.8 Q(h,s)] + 0.6 x[3+0.8 x [Q(l,s)]]$$

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

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



Policy deterministica 2



$$\alpha=0.4, \beta=0.1, \gamma=0.8, \\ R^{\text{search}}=3, R^{\text{wait}}=1, R^{\text{dead}}=-3, R^{\text{auto}}=0$$

s = High - a = Search;
s = Low - a = Search;

$$Q(h,s)[1-0.32] = 1.2 + 1.8 + 0.48 \times Q(l,s) \\ Q(l,s)[1-0.08] = 0.3 - 2.7 + 0.72 Q(h,s)$$

$$Q(h,s) [0.68] - 0.48 Q(l,s) = 3. \\ Q(l,s) [0.92] - 0.72 Q(h,s) = -2.4$$

$$\rightarrow Q(h,s) = 5.7429 \\ \rightarrow Q(l,s) = 1.8857$$

La policy è migliore per entrambe le coppie stato-azione.

A.A. 2024-2025

53/55

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



Osservazioni



Posso migliorare l'azione in uno stato (e.g. nello stato High):

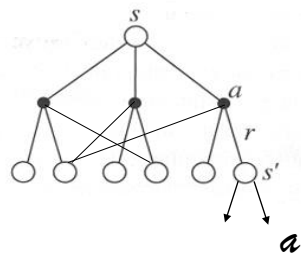
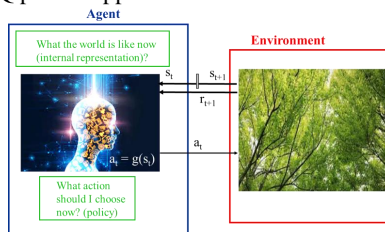
$$\begin{array}{ll} s = \text{High} \rightarrow a = \text{Wait}; & Q(h,w) = 5 \\ s = \text{Low} \rightarrow a = \text{Search}; & Q(l,s) = 1,304 \end{array} \quad \begin{array}{l} a = \text{Search} \\ Q(h,s) = 5.7429 \end{array}$$

Questo ha impatto anche sugli altri stati, sullo stato low →

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

E' un sistema interconnesso in cui l'azione in uno stato si ripercuote a valle, anche sugli stati futuri (e le azioni scelte da lì in poi).

La scelta di un'azione o di un'altra, cambia la funzione Q per le coppie stato-azione future.



A.A. 2024-2025

54/55

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



Sommario



Il Reinforcement Learning.

Processi Markoviani.

Esempi di calcolo