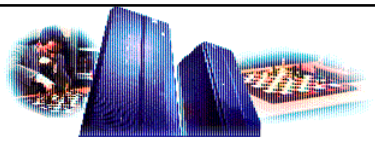
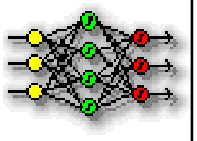
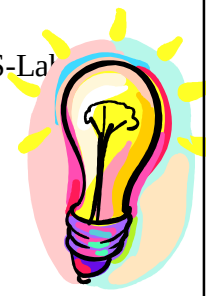




# Sistemi Intelligenti
# Reinforcement Learning:
# Iterative policy evaluation


Alberto Borghese

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



## Sommario

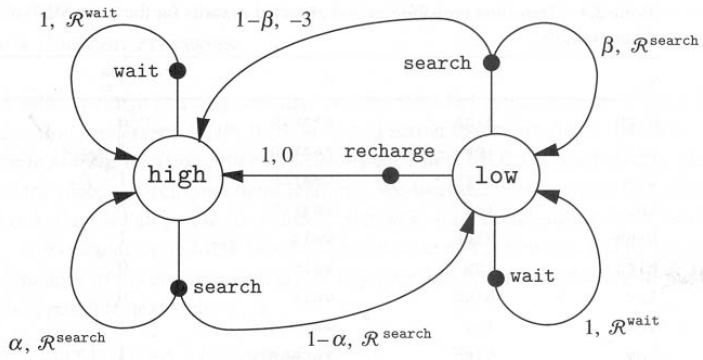
Le equazioni di Bellman

Stima iterativa della funzione valore

# Esempio di calcolo della Value function

Value function
Q(high, search) = ?
Q(low, wait) = ?

Chosen Policy
a(high) = search
a(low) = wait






# Esempio di calcolo di Q

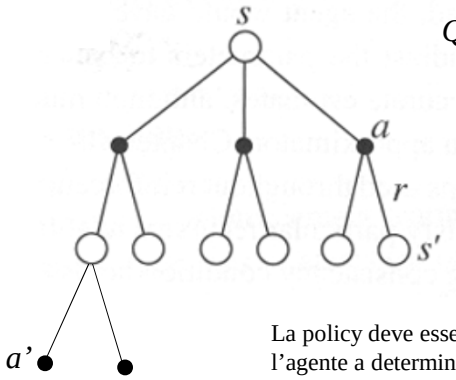
Con policy deterministica:

- $Q(h,s) = \alpha [R^S + \gamma\ Q(h,w)] + (1- \alpha) \times [R^S + \gamma\ Q(h,l)]$
- $Q(l,s) = \beta \times [R^s + \gamma\ \times Q(l,s)] + (1- \beta) \times [R^D + \gamma\ Q(h,w)]$

Con policy stocastica:

- $Q(h,s) = \alpha [R^S + \gamma\ [\varepsilon\ Q(h,w) + (1- \varepsilon\ ) Q(h,s)] +$
  $(1- \alpha) \times [R^S + \gamma\ Q(h,l)]$
- $Q(l,s) = \beta \times [R^s + \gamma\ \times [\theta\ Q(l,w) + \eta\ Q(l,s)] + (1 - \eta - \theta) Q(l, r)] ] +$
  $(1- \beta) \times [R^D + \gamma\ [\varepsilon\ Q(h,w) + (1- \varepsilon\ ) Q(h,s)]]$

# Analisi ad un passo




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

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).

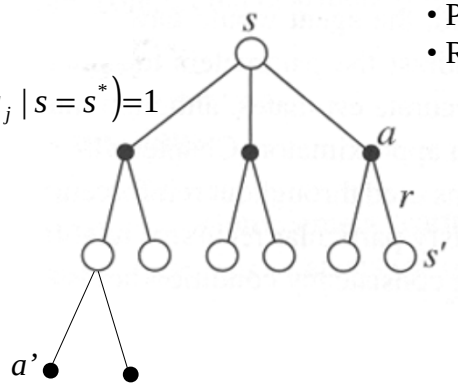
Archi multipli fuoriuscenti da uno stato, sono associati alla policy.



# Value function e modelli markoviani

Anche la policy può essere stocastica.

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




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

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

L'azione scelta in s' può essere scelta in modo stocastico (e.g. ε-greedy policy)

# Il modello markoviano

Il comportamento dell'ambiente è definito dallo stato: S = $\{s_j\}$
Per ogni stato l'agente sceglie un'azione: a = a(s)  A = $\{a_k\}$
**Policy di un agente: $\pi$(s, a) è quanto dobbiamo definire.**

L'ambiente ha una evoluzione stocastica rappresentata da un MDP:

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

Inoltre, ad ogni istante fornisce un reward immediato associato alla transizione, stimato all'istante t come:

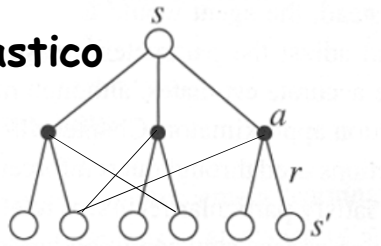
$$R_{s_t=s->s_{t+1}=s'|a_t=a} = E\{r_{t+1} = r' | s_t = s, a_t = a, s_{t+1} = s'\}$$

$\forall s \in S; \forall a \in A$



# Reward stocastico

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




reward = 3 – stocastico (da una distribuzione statistica)

È in realtà un valore condizionato in s e vale:

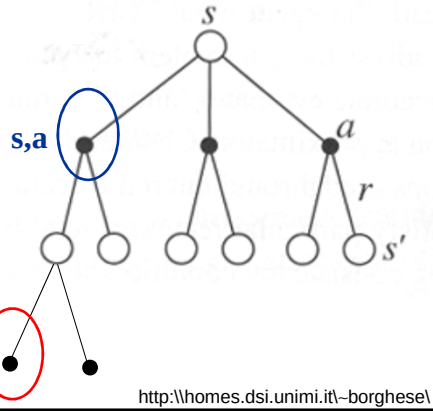
Pr(reward = r | s) = Pr(reward = r | s') * Pr(a | s) * Pr(s' | s,a)
Questa è la probabilità congiunta di stato prossimo, azione e reward.

## Calcolo ricorsivo della Value function

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

$$Q^{\pi}(s',a')=E_{\pi}\{R_{t+1} \mid s_{t+1}=s', a_{t+1}=a'\}$$

Relazione?






## Calcolo ricorsivo della Value function

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

Isolo il reward ad un passo nella serie dei reward.

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

$$Q^{\pi}(s_t,a_t)=E_{\pi}\left\{r_{t+1}+\sum_{k=0}^{\infty}\gamma^{k+1} r_{t+k+2} \,\middle|\, s_t=s, a_t=a\right\}$$

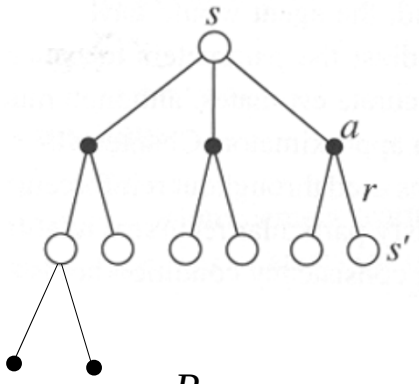
Io termine

IIo termine

# $Q^{\pi}(s,a)$ : primo termine

$$E_{\pi}\{r_{t+1} \mid s_t = s, a_t = a\} = \sum_{s'} P_{s \to s' | a_j} \left[ R_{s \to s' | a_j} \right]$$

Per ogni stato devo valutare:
- Più stati prossimi
- Reward stocastici nella transizione ad un passo




**Visione Statistica**: Probabilità di ottenere il reward: $R_{s \to s' | a_j}$ condizionata all'arrivare nello stato s'.



# $Q^{\pi}(s,a)$ : secondo termine

$$Q^{\pi}(s_t, a_t) = E_{\pi}\left\{ \sum_{k=0}^{\infty} \gamma^{k+1} r_{t+k+2} \,\middle|\, s_t = s, a_t = a \right\} =$$

$$\gamma E_{\pi}\left\{ \sum_{k=0}^{\infty} \gamma^{k} r_{t+k+2} \,\middle|\, s_t = s, a_t = a \right\} =$$

$$\gamma \sum_{s'} \left( \Pr(s_{t+1} = s' \mid s_t = s, a_t = a) E_{\pi}\left( \sum_{k=0}^{\infty} \gamma^{k} r_{t+k+2} \mid s_{t+1} = s' \right) \right) =$$

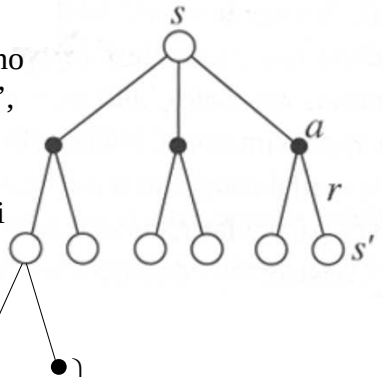
$$\gamma \sum_{s'} \left( \Pr(s_{t+1} = s' \mid s_t = s, a_t = a) \right)$$

$$\left( \sum_{a'} \Pr(a_{t+1} = a' \mid s_{t+1} = s') E_{\pi}\left( \sum_{k=0}^{\infty} \gamma^{k} r_{t+k+2} \mid s_{t+1} = s', a_{t+1} = a' \right) \right) =$$

$$\gamma \sum_{s'} \left( \Pr(s_{t+1} = s' \mid s_t = s, a_t = a) \right) \left( \sum_{a'} \Pr(a_{t+1} = a' \mid s_{t+1} = s') Q(s', a') \right)$$

# $Q^\pi(s,a)$ : secondo termine

In (s,a) confluiranno i reward a lungo termine di tutti gli stati prossimi, s', ciascuno pesato con la probabilità di passare da s a s', ovverosia, in termini statistici, condizionati alla realizzazione della transizione di stato, s-> s' e dai reward a lungo termine, ottenuti scegliendo in s' l'azione a'.




$$Q^\pi(s_t,a_t)=E_\pi\left\{\sum_{k=0}^{\infty}\gamma^{k+1}r_{t+k+2}\,\middle|\,s_t=s,a_t=a\right\}=$$

$$\gamma\sum_{s'}\left(\Pr(s_{t+1}=s'|\,s_t=s,a_t=a)\right)\left(\sum_{a'}\Pr(a_{t+1}=a'|\,s_{t+1}=s')Q(s',a')\right)$$



# Calcolo ricorsivo della Value function

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

$$Q^\pi(s',a')=E_\pi\{R_{t+1}\,|\,s_{t+1}=s',a_{t+1}=a'\}$$

Legame?

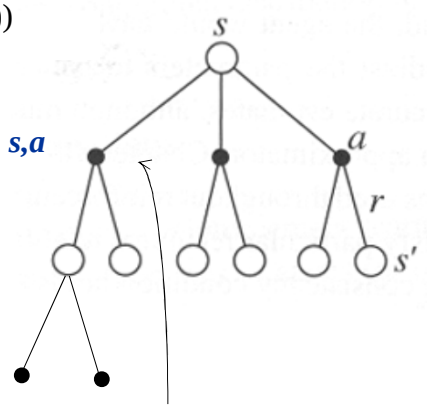
Bellman's equation

Next-state Policy

$$Q^\pi(s,a)=\left\{\sum_{s_l'}\left\{P_{s\to s_l'|a}\left[R_{s\to s_l'|a}+\gamma\sum_{a'_j}\pi(a'_j,s_l')Q^\pi(s_l',a_j')\right]\right\}\right\}$$

# Osservazioni

$Q^{\pi}(s,a) = funz(Q^{\pi}(s',a'))$

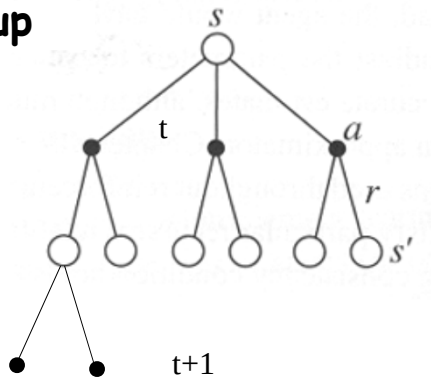



**Backwards in time**

$$Q^{\pi}(s,a)=\left\{\sum_{s_l'}\left\{P_{s\to s_l'|a}\left[R_{s\to s_l'|a}+\gamma\sum_{a'_j}\pi(a'_j,s_l')Q^{\pi}(s_l',a_j')\right]\right\}\right\}$$



# Tecnica full-backup




Back-up

$\pi(s,a)$ fissata

Conosciamo $Q_k(s_t,a_t)$ $\forall s_t$, anche per $s'_{t+1}$ quindi:
Analizziamo la transizione da $s_t,a_t$ -> $(s'_{t+1}, a'_{t+1})$
Calcoliamo un nuovo valore di Q per s,a: $Q_{k+1}(s_t,a_t)$ congruente con:
$Q_k(s_t,a_t)$ ed $r_{t+1}$
*Full backup* se esaminiamo tutti gli s' (cf. DP).
Da s' mi guardo indietro ed aggiorno Q(s, a).

$\pi$ fissata

# Sommario

Le equazioni di Bellman

Stima iterativa della funzione valore



# Calcolo iterativo della Value Function

Per ogni stato s, estratto a caso, analizziamo una singola transizione.

Equazione di Bellman per "iterative policy evaluation":

$$Q_{k+1}^{\pi}(s,a)=\left\{\sum_{s_l'}\left\{P_{s\to s_l'|a}\left[R_{s\to s_l'|a}+\gamma\sum_{a'_j}\pi(a'_j,s_l')Q_k^{\pi}(s_l',a_j')\right]\right\}\right\}$$

Mi fido di $Q_{k+1}(s',a')$ (Backup)

$$\lim_{k\to\infty}\{Q_k(s,a)\}=Q^{\pi}(s,a)$$

# Iterative policy evaluation

Evoluzione del sistema da s(t=0) a {s'(t = T)} utilizzando la policy $\pi(s,a)$, prefissata.

Quanto valgono gli stati?

Parto da $Q(s(t=0), a(s(t=0))_{k=0}$ arbitraria, otterrò una value function per ogni stato-azione che sarà funzione di Q(s(t=0), a(s(t=0)).

Devo migliorare, come?

Utilizziamo l'informazione sul **passato**, tenendo conto che gli stati sono in numero finito e vengono ri-visitati.

$\{Q\}^0$, $\{Q\}^1$, $\{Q\}^2$, $\{Q\}^3$, $\{Q\}^4$, $\{Q\}^5$, ……. $\{Q\}^\infty$

$$\lim_{k\to\infty}\{Q_k(s,a)\} = Q^\pi(s,a)$$



# Fondamenti del metodo

• Supponiamo di essere all'istante t. In questo istante t, si può passare ad un certo insieme di stati: $\{s'_{t+1}\}$.

•Analizziamo un solo passo: cosa succede nella transizione da t a t+1.

• Migliorare la stima della nostra Value Function ad ogni iterazione.

# Algoritmo per "iterative policy evaluation", versione batch

Partiamo da una politica π(s,a) data.
Definiamo una soglia di convergenza τ
Inizializziamo V(s) = 0 ∀s, compreso gli stati finali.
Repeat

Forward pass

{ Δ = 0;
for s = 1 : N // ∀s, ≠ TS
{ for a = 1: M // ∀a that can be chosen in s
{ $W(s,a) = \sum_{s'} P^{a}_{s->s'} \left[ R^{a}_{s->s'} + \gamma \sum_{a_j} \pi(s,a_j) Q(s',a'_j) \right]$ // W(s) è $Q_{k+1}(s,a)$
Δ = max(Δ, | Q(s,a) – W(s,a) |)
}
}
for s=1:N; a = 1:M
Q(s,a) = W(s,a);
} Until (Δ < τ);



# Interpretazione dell'update (batch o trial)

$$Q(s,a) = \sum_{s'} P^{a}_{s->s'} \left[ R^{a}_{s->s'} + \gamma \sum_{a_j} \pi(s,a_j) Q(s',a'_j) \right]$$

Al termine dell'aggiornamento dei Q(s,a) per tutti gli stati-azioni, $Q(s,a) = Q_{new}(s,a)$. **Aggiornamento batch**.

Utilizzerò in parte già il nuovo valore di Q(s,a) all'interno dell'equazione di aggiornamento. **Aggiornamento per trial.**

Entrambe le modalità di aggiornamento convergono.

# Algoritmo per "iterative policy evaluation", versione per trial

Partiamo da una politica π(s,a) data.
Definiamo una soglia di convergenza τ
Inizializziamo V(s) = 0 ∀s, compreso gli stati finali.
Repeat
{ Δ = 0;
for s = 1 : N // ∀s, ≠ TS
{ for a = 1: M // ∀a that can be chosen in s
{ Value = $Q_k(s,a)$
$Q_{k+1}(s,a) = \sum_{s'} P^a_{s->s'} \left[ R^a_{s->s'} + \gamma \sum_{a'} \pi(s,a_j) Q_k(s',a'_j) \right]$
Δ = max(Δ, | $Q_{k+1}(s,a)$ – Value|)
}
}
} Until (Δ < τ);

Forward pass



# Problematiche legate al calcolo di Q(s,a): problema di policy evaluation

3 assunzioni:

1) Conoscenza della dinamica dell'ambiente: P(s->s'| a)
2) Conoscenza della policy (eventualmente stocastica), π(s, a)
3) Potenza di calcolo sufficiente
4) Proprietà Markoviane dell'ambiente (definizione di uno stato).

Le equazioni contengono dei termini statistici (valori attesi).

Soluzione di un sistema lineare in N incognite (numero di stati).

Come mai posso determinare la Value function per la policy π(.), se questa si basa sul reward che riceverò negli istanti futuri?

C'e' poca interazione con l'ambiente e molta simulazione (cf. metodi Montecarlo).

# Riassunto

Posso determinare la Value function in modo ricorsivo. Per ogni stato, sarà funzione dell'output dell'ambiente in quell'istante (attraverso la funzione stato prossimo ed il reward istantaneo) e della policy scelta in quell'istante e dei reward a lungo termine attesi negli stati in cui l'ambiente mi porta.

Per scegliere la policy devo esaminare il reward a lungo termine che mi si prospetta nello stato in cui mi trovo e scegliere l'azione che lo massimizza.



# Problematiche legate al calcolo di Q*(s,a)

Soluzione vicina alla ricerca esaustiva. Devo valutare per ogni stato tutte le possibili azioni (devo trovare il massimo).

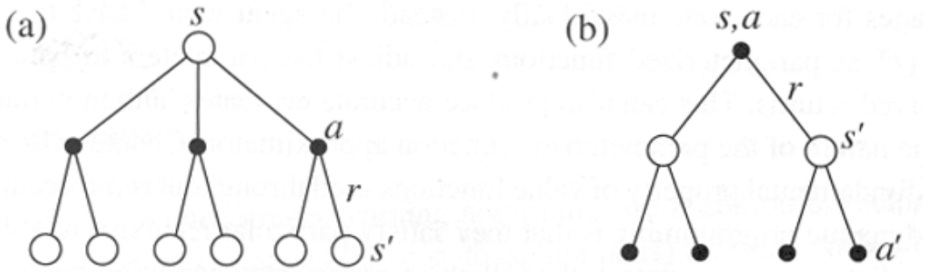
Per tutte le possibili azioni devo calcolare la probabilità di transizione allo stato successivo e di ottenere una certa reward.

3 assunzioni:

1) Conoscenza della dinamica dell'ambiente: P(s->s'| $a_j$)
2) Potenza di calcolo sufficiente
3) Proprietà Markoviane dell'ambiente (definizione di uno stato).

Soluzioni approssimate.

# Policy



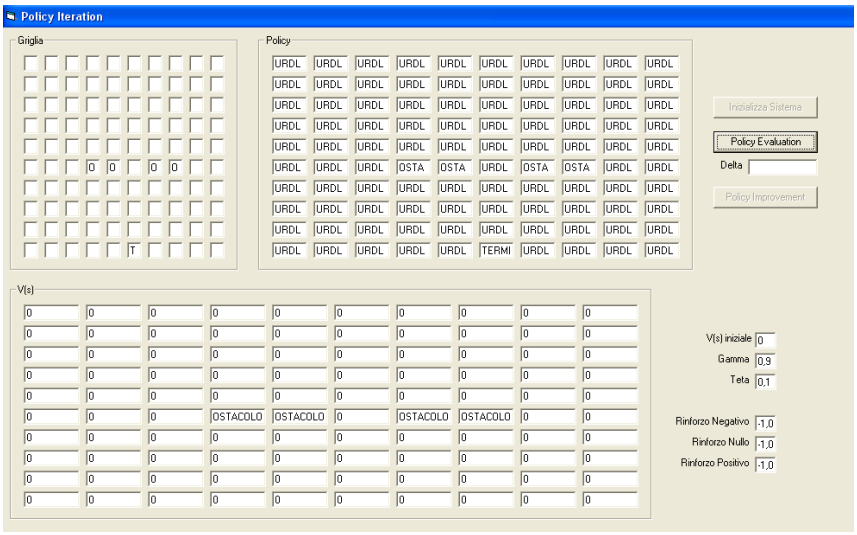

La policy deve essere ancora determinata. Come fa l'agente a determinare la policy ottimale?

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.



# Iterative policy evaluation




Sommaruga_PolicyIteration_VisualBasic_Labirinto

# Sommario

Le equazioni di Bellman

Stima iterativa della funzione valore

Osservazioni sulla funzione valore ottima