

Sistemi Intelligenti Reinforcement Learning: Iterative policy evaluation

Alberto Borghese

Università degli Studi di Milano
Laboratorio di Sistemi Intelligenti Applicati (AIS-La)
Dipartimento di Informatica
borgnese@di.unimi.it



A.A. 2019-2020

1/38

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



Sommario



Le equazioni di Bellman

Stima iterativa della funzione valore

Policy iteration

A.A. 2019-2020

2/38

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

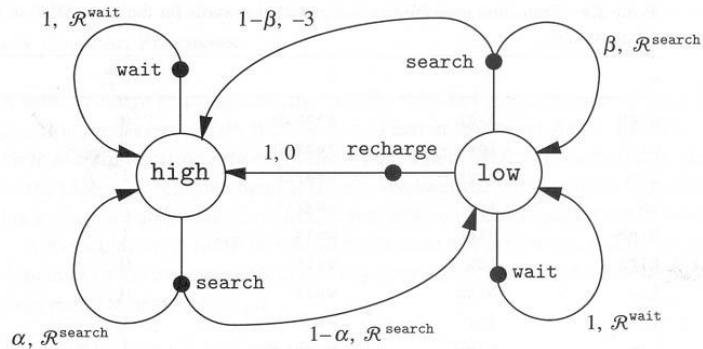


Esempio di calcolo della Value function

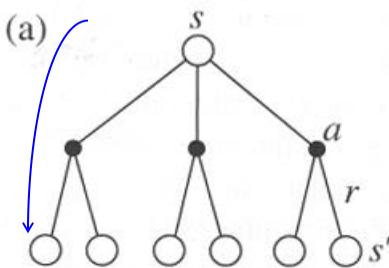


Value function
 $V(\text{high}) = ?$
 $V(\text{low}) = ?$

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



Analisi ad un passo



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

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.

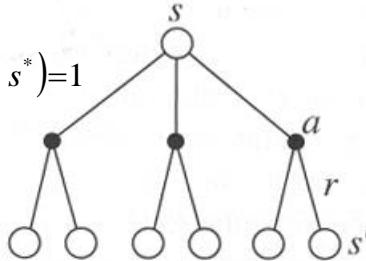


Value function e modelli markoviani



Anche la policy può essere stocastica.

$$\sum_{j=1}^{N_{\text{azioni}}} \Pr(a_j | s = s^*) = 1$$



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

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

Per ogni stato devo valutare:

- Più azioni
- Per ogni azione, più stati prossimi
- Reward stocastici.



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 può definire (e ottimizzare) l'agente.

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

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

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

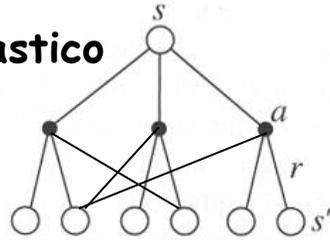
$$R_{s_t=s \rightarrow s_{t+1}=s' | a_t=a} = E\{r_{t+1} = r' | s_t = s, a_t = a, s_{t+1} = s'\}$$

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



Reward stocastico

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



Reward stocastico (da una distribuzione statistica)

È un valore condizionato a s , a , s' e vale:

$$\Pr(\text{reward} = r' | s, a, s') = \Pr(\text{reward} = r' | s') * \Pr(a | s) * \Pr(s' | s, a)$$

Questa è la probabilità condizionata a: stato prossimo, azione e reward.



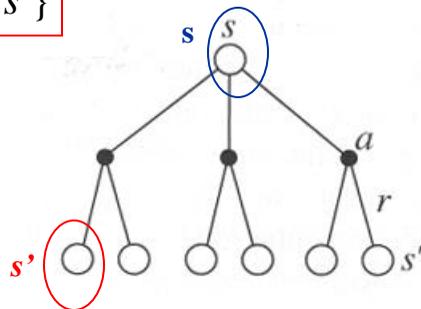
Calcolo ricorsivo della Value function



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

$$V^\pi(s') = E_\pi\{R_{t+1} | s_{t+1} = s'\}$$

Relazione tra $V^\pi(s)$ e $V^\pi(s')$?





Calcolo ricorsivo della Value function



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

Isolo il reward ad un passo nella serie dei reward.

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

$$V^\pi(s_t) = E_\pi \left\{ \left(r_{t+1} + \sum_{k=0}^{\infty} \gamma^{k+1} r_{t+k+2} \right) \mid s_t = s \right\}$$

Io termine

Illo termine



$V^\pi(s)$: primo termine

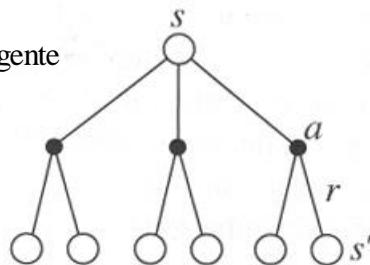


$$E_\pi \{ r_{t+1} \mid s_t = s \} = \sum_{a_j} \pi(s, a_j) \left[\sum_{s'} P_{s \rightarrow s' \mid a_j} R_{s \rightarrow s' \mid a_j} \right]$$

Per ogni stato devo valutare:

- Più azioni
- Più stati prossimi
- Reward stocastici nella transizione ad un passo

Agente



Visione Statistica: Probabilità di ottenere il reward: condizionata all'arrivare nello stato s' .

$$R_{s \rightarrow s' \mid a_j}$$



$V^\pi(s)$: secondo termine



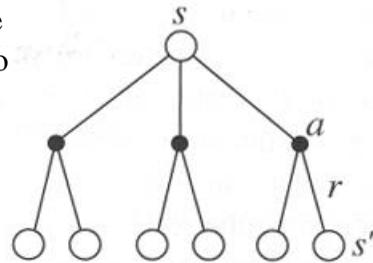
$$\begin{aligned}
 V^\pi(s_t) &= E_\pi \left\{ \sum_{k=0}^{\infty} \gamma^{k+1} r_{t+k+2} \mid s_t = s \right\} = \\
 &\gamma E_\pi \left\{ \sum_{k=0}^{\infty} \gamma^k r_{t+k+2} \mid s_t = s \right\} = \\
 &\gamma \left[\sum_j \Pr(a_j = a \mid s_t = s) \sum_{s'} \Pr(s_{t+1} = s' \mid s_t = s, a_t = a_j) \right] E_\pi \left(\sum_{k=0}^{\infty} \gamma^k r_{t+k+2} \mid s_{t+1} = s' \right) \\
 &= \\
 &= \gamma \underbrace{\sum_j \Pr(a_j = a \mid s_t = s) \sum_{s'} \left(\Pr(s_{t+1} = s' \mid s_t = s, a_t = a_j) V^\pi(s') \right)}_{\Pr(s_{t+1} = s' \mid s_t = s)}
 \end{aligned}$$



$V^\pi(s)$: secondo termine



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



$$\begin{aligned}
 V^\pi(s_t) &= E_\pi \left\{ \sum_{k=0}^{\infty} \gamma^{k+1} r_{t+k+1} \mid s_t = s \right\} = \\
 &\sum_j \left(\Pr(a_t = a \mid s_t = s) \right. \\
 &\quad \left. \sum_{s'} \left(\Pr(s_{t+1} = s' \mid s_t = s, a_t = a) \right) \left(\Pr(r_{t+1} = r' \mid s_t = s, a_t = a, s_{t+1} = s') + \gamma V^\pi(s') \right) \right)
 \end{aligned}$$



Calcolo ricorsivo della Value function



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

$$V^\pi(s') = E_\pi \{R_{t+1} | s_{t+1} = s'\}$$

Legame?

Bellman's equation

Policy

Next-state

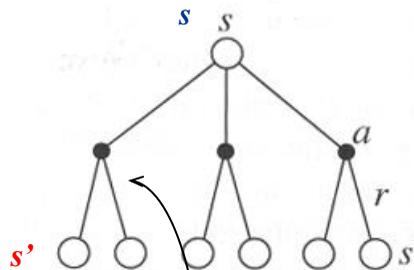
$$V^\pi(s) = \sum_j \pi(a_j, s) \left\{ \sum_{s_l'} \left\{ P_{s \rightarrow s_l' | a_j} \left[R_{s \rightarrow s_l' | a_j} + \gamma V^\pi(s_l') \right] \right\} \right\}$$



Osservazioni



Backwards in time

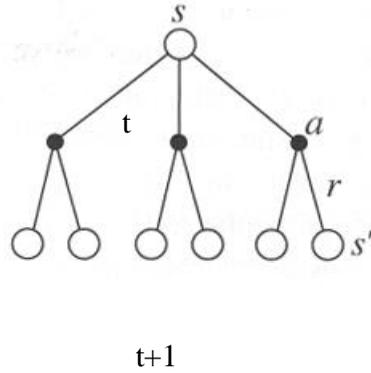


$$V^\pi(s) = \sum_j \pi(a_j, s) \left\{ \sum_{s_l'} \left\{ P_{s \rightarrow s_l' | a_j} \left[R_{s \rightarrow s_l' | a_j} + \gamma V^\pi(s_l') \right] \right\} \right\}$$



Tecnica full-backup

Back-up ↑



$\pi(s,a)$ fissata

t+1

Conosciamo $V_k(s_t) \forall s_t$, anche per s'_{t+1} quindi:

Analizziamo la transizione da $s_t, a_t \rightarrow (s'_{t+1})$

Calcoliamo un nuovo valore di V per s : $V_{k+1}(s_t)$ congruente con:

$V_k(s_t)$ ed r_{t+1}

Full backup se esaminiamo tutti gli s' (cf. DP).

Da s' mi guardo indietro ed aggiorno $V(s)$.

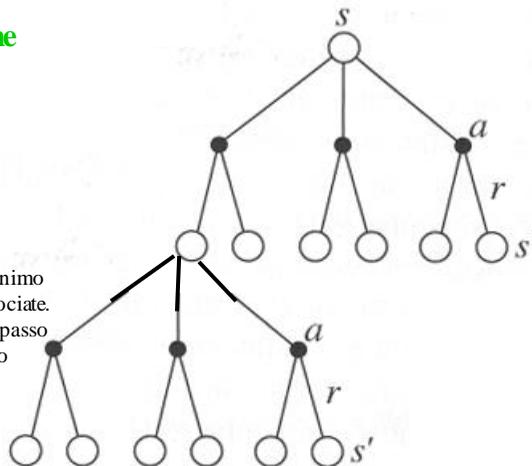
π fissata



Analogia con la programmazione dinamica



Backwards in time



Per ogni stato calcolo il costo minimo per arrivare in s e il percorso associato. Il calcolo è a 1 passo: costo di 1 passo + costo totale dallo stato di arrivo



Sommario



Le equazioni di Bellman

Stima iterativa della funzione valore

Policy iteration



Calcolo iterativo della Value Function



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

Equazione di Bellman per “iterative policy evaluation”:

$$V_{k+1}^{\pi}(s) = \sum_j \pi(a_j, s) \left\{ \sum_{s'} P_{s \rightarrow s' | a_j} \left[R_{s \rightarrow s' | a_j} + \gamma V_k(s') \right] \right\}$$

Mi fido di $V_k(s')$ (Backup)

$$\lim_{k \rightarrow \infty} \{V_k(s)\} = V^{\pi}(s)$$



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

Devo migliorare, come?

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

$\{V\}^0, \{V\}^1, \{V\}^2, \{V\}^3, \{V\}^4, \{V\}^5, \dots, \{V\}^\infty$

$$\lim_{k \rightarrow \infty} \{V_k(s)\} = V^\pi(s)$$



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 $\pi(s,a)$ data.

Definiamo una soglia di convergenza τ

Inizializziamo $V(s) = 0 \forall s$, compreso gli stati finali.

Repeat

```

{  Δ = 0;
  for s = 1 : NS           // ∀s, ≠ TS
  {  Temp_V(s) = 0;
    for a = 1: NA         // ∀a that can be chosen in s
    {  Pr_a = policy(s,a);
      for snext = 1 : NS
      {  PrSnext = NextState(s,a);
        reward = ComputeReward(s,a,snext);
        Temp_V(s) = Pr_a*PrSnext*(reward + γV(snext);
      } } }
    for s=1:NS;
    {  if ( | Temp_V(s) - V(s) | > Δ )
      {  Δ = | Temp_V(s) - V(s) |;
        V(s) = Temp_V(s);
      }
    }
  }
} Until (Δ < τ);

```

Forwards
pass

Backwards
pass



Interpretazione dell'update (batch o trial)



$$V(s) = \sum_{a_j} \pi(s, a_j) \sum_{s'} P_{s \rightarrow s'}^{a_j} [R_{s \rightarrow s'}^{a_j} + \gamma V(s')]$$

Al termine dell'aggiornamento dei $V(s)$ per tutti gli stati, $V(s) = V_{\text{new}}(s)$. **Aggiornamento batch.**

Utilizzerò in parte già il nuovo valore di $V(s)$ 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 $\pi(s,a)$ data.

Definiamo una soglia **relativa** di convergenza τ

Inizializziamo $V(s) = 0 \forall s$, compreso gli stati finali.

Repeat

{ $\Delta = 0$;

for $s = 1 : N$ // $\forall s, \neq TS$

{ Value = $V(s)$;

$$V_{k+1}(s) = \sum_{a_j} \pi(s, a_j) \sum_{s'} P_{s \rightarrow s'}^{a_j} [R_{s \rightarrow s'}^{a_j} + \gamma V(s')]$$

$$\Delta = \max(\Delta, (| \text{Value} - V_{k+1}(s) |) / | \text{Value} |)$$

}

} Until ($\Delta < \tau$);



Problematiche legate al calcolo di $V(s)$: problema di policy evaluation



3 assunzioni:

- 1) Conoscenza della dinamica dell'ambiente: $P(s \rightarrow s' | a)$
- 2) Conoscenza della policy (eventualmente stocastica), $\pi(s, a)$
- 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 $\pi(\cdot)$, se questa si basa sul reward che riceverò negli istanti futuri?

C'è 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 $V^*(s)$



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: $P(s \rightarrow s' | a_j)$
- 2) Potenza di calcolo sufficiente
- 3) Proprietà Markoviane dell'ambiente (definizione di uno stato).

Soluzioni approssimate.



Iterative policy evaluation



Reinforcement Learning

File Help

Simulazione

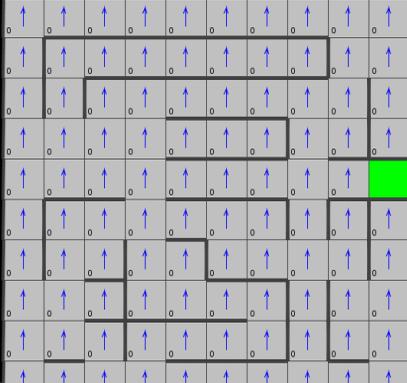
PARAMETRI

Discount:

Convergi:

Console

Log: Griglia con Penalità muro:10 e Reward Goal +20 | Value Iteration



Universita degli studi di Milano, Corso Di Sistemi Intelligenti 2008/2009



DeRosaRocco_ValueIteration

A.A. 2019-2020
27/38
<http://borgese.di.unimi.it/>



Sommaro



Le equazioni di Bellman

Stima iterativa della funzione valore

Policy iteration

A.A. 2019-2020
28/38
<http://borgese.di.unimi.it/>



Miglioramento della policy



Tutti gli stati sono valutati in funzione di una policy data.

Condizioni di funzionamento dell'agente:

- Policy **deterministica**: $a = \pi(s)$.
- Ambiente **stocastico**.

Cosa succede se cambiamo la policy per un certo stato s_m ? $a_{new} \neq \pi(s_m)$.
Cosa viene influenzato?

Scelgo a_{new} in s_m , visiterò una certa sequenza di stati, per questi stati seguirò la policy precedente per $s \neq s_m$. Cosa viene influenzato?

Come faccio a valutare se miglioro la policy o no?

$$Q^\pi(s, a) = V^\pi(s) \text{ se } a = \pi(s)$$



Effetto del cambiamento della policy



Cambia, a, cambiano i possibili stati successivi ad s_m , $\{s_{t+k}\}$, ed il reward a lungo termine:

$$Q^\pi(s_m, a_{new}) = E_\pi \{ r_{t+1} + \gamma V^\pi(s_{t+1}) \mid s_t = s_m, a_t = a_{new} \neq \pi(s_m) \} =$$

$$\sum_{s'} P_{s_m \rightarrow s'}^{a_{new}} [R_{s_m \rightarrow s'}^{a_{new}} + \gamma V^\pi(s')] \quad V(s) = \text{value function sullo stato}$$

?

$$Q^\pi(s_m, a_{new}) \geq Q^\pi(s_m, a = \pi(s_m)) \quad \forall s, a?$$

Se il reward fosse migliore con a_{new} , sceglierò sempre a_{new} in s_m .

Il reward a lungo termine può essere maggiore (minore) solamente se aumenta (diminuisce) il reward totale "visto" ad un passo (reward del passo + reward successivo).



Enunciato del teorema del miglioramento della policy

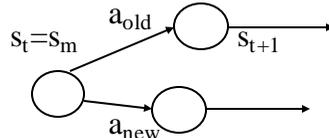


$$Q^{\pi}(s, a) = \sum_k P_{s \rightarrow s_k | a} [R_{s \rightarrow s_k | a} + \gamma V^{\pi}(s_k)]$$

Ipotesi: π and π' deterministic policies
 $Q^{\pi}(s_m, \pi'(s_m)) \geq V^{\pi}(s_m)$

$$Q^{\pi}(s, a_{new} = \pi'(s_m)) = \sum_k P_{s_m \rightarrow s_k | a_{new}} [R_{s_m \rightarrow s_k | a_{new}} + \gamma V^{\pi}(s_k)]$$

Tesi: π' è meglio di π . Cioè: $V^{\pi'}(s) \geq V^{\pi}(s) \forall s$.
 $Q^{\pi'}(s, a_{new}) \geq Q^{\pi}(s, a_{old})$



Dimostrazione del teorema del miglioramento della policy



Analizziamo la seguente condizione:

$\pi' = \pi \forall s$ tranne che per s_m per il quale si applica l'azione:
 $a_{new} = \pi'(s_m)$

Risulta che il reward a lungo termine è maggiore per $a_{new} = \pi'(s)$.

$$V^{\pi'}(s) = Q^{\pi'}(s, a_{new} = \pi'(s)) \geq Q^{\pi}(s, a = \pi(s)) = V^{\pi}(s)$$

Tesi: π' è meglio di π . Cioè: $V^{\pi'}(s) \geq V^{\pi}(s) \forall s$ (ed in particolare per gli altri stati s)



Dimostrazione del teorema del miglioramento della policy



Hp: $Q^\pi(s, \pi'(s)) \geq V^\pi(s) \quad \forall s \quad \pi'(s, a)$ è migliore per almeno uno stato

$$V^\pi(s) \leq Q^\pi(s, \pi'(s))$$

$$= E_{\pi'}\{r_{t+1} + \gamma V^\pi(s_{t+1}) \mid s_t = s\}$$

$$\leq E_{\pi'}\{r_{t+1} + \gamma Q^\pi(s_{t+1}, \pi'(s_{t+1})) \mid s_t = s\}$$

$$\leq E_{\pi'}\{r_{t+1} + \gamma E_{\pi'}(r_{t+2} + \gamma V^\pi(s_{t+2})) \mid s_t = s\}$$

$$= E_{\pi'}\{r_{t+1} + \gamma r_{t+2} + \gamma^2 V^\pi(s_{t+2}) \mid s_t = s\}$$

Sostituisco ancora $Q^{\pi^*}(\cdot)$

$$\leq E_{\pi'}\{r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \dots \mid s_t = s\}$$

Th: $V^\pi(s) \leq V^{\pi^*}(s)$



Osservazioni



$$s = s_m \quad Q^\pi(s_m, \pi'(s)) \geq Q^\pi(s_m, \pi(s))$$

$$s \neq s_m \quad Q^\pi(s, a) = E_{\pi'}\{r_{t+1} + \gamma V^\pi(s_{t+1}) \mid s_t = s\}$$

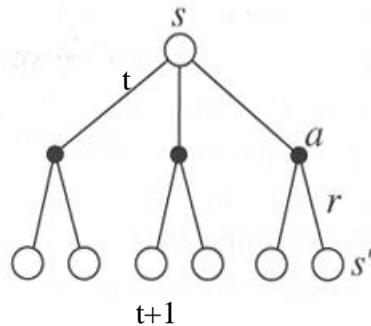
$$= E_{\pi'}\{r_{t+1} + \gamma Q^\pi(s_{t+1}, \pi(s_{t+1})) \mid s_t = s\}$$

Se $s_{t+k} = s_m$ migliore la $Q(s, a)$.

Se nessun $s_{t+k} = s_m$. Non varia la $Q(s, a)$.



Visione grafica del miglioramento



Ogni volta che sono in uno stato, s , scelgo un'azione che migliora il reward a lungo termine ottenuto da quell'istante/stato in poi.

Per gli altri stati, il reward a lungo termine non viene modificato ogni volta che l'albero uscente da s passa per s .



Ottimizzazione policy



Per ogni stato scelgo le azioni secondo la policy: $\pi(s,a)$.

Posso ordinare la Value function $Q(s,a)$ in ordine decrescente, in funzione delle azioni scelte in s (policy).

Si definisce una policy, π_1 , migliore di un'altra, π_2 , se e solo se:

$$Q^{\pi_1}(s,a(s)) \geq Q^{\pi_2}(s,a(s)) \quad \forall s.$$

In particolare si definisce una politica ottima, π^* , se e solo se:

$$Q^*(s,a(s)) \geq V^{\pi}(s,a(s)) \quad \forall s$$

$$Q^*(s,a(s)) \geq Q^{\pi}(s,a(s)) \quad \forall [s,a]$$



Policy iteration



Iterazione tra:

- Calcolo iterativo della Value function (iterative policy evaluation)
- Miglioramento della policy (policy improvement)

$$\begin{array}{ccccccccccc} \pi_0 & \rightarrow & V^{\pi_0} & \rightarrow & \pi_1 & \rightarrow & V^{\pi_1} & \rightarrow & \pi_2 & \rightarrow & V^{\pi_2} & \rightarrow & \dots \\ & & & & \rightarrow & & \rightarrow & & & & & & \end{array}$$

Converge velocemente ad una buona politica
(cf. Software Sommaruga)



Sommario



Le equazioni di Bellman

Stima iterativa della funzione valore

Policy iteration