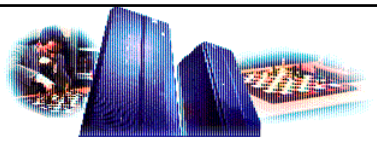
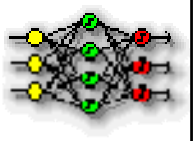




Sistemi Intelligenti
Reinforcement Learning: Iterative policy evaluation

Alberto Borghese

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

Sommario

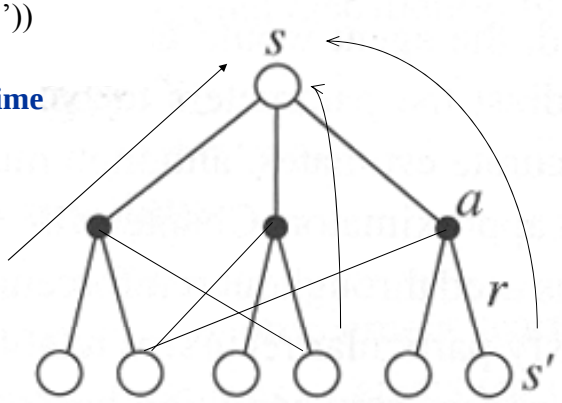
Determinazione ricorsiva della Value function

Determinazione della policy ottima.

Osservazioni

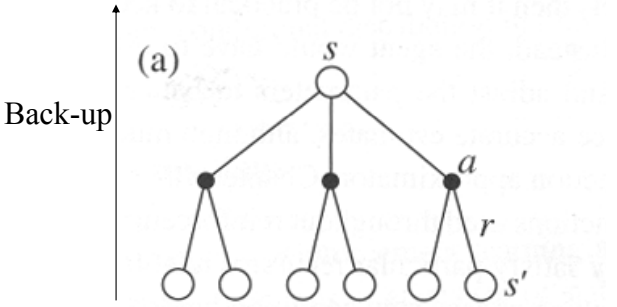
$V^{\pi}(s) = funz(V^{\pi}(s'))$

Backwards in time



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

Tecnica full-backup



Conosciamo $V_k(s(t))$ ∀s, anche per s' quindi
Analizziamo la transizione da s(t) -> {s'(t+1)}
Calcoliamo un nuovo valore di 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).

π fissata

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

Calcolo ricorsivo della Value function

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

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

Legame?

Policy Next-state

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

$$V^{\pi}(s) = \sum_{a_j} \pi(a_j, s) \sum_{s'} P_{s \to s'|a_j} R_{s \to s'|a_j} + E_{\pi}\left\{\gamma \sum_{k=0}^{\infty} \gamma^k r_{t+k+2} \,\middle|\, s_t = s\right\}$$

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

Bellman's equation

Calcolo iterativo della Value Function

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

Equazione di Bellman per "iterative policy evaluation":

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

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

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

Iterative policy evaluation

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

Quanto valgono gli stati?

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

Devo migliorare, come?

Utilizziamo l'informazione sul **passato**.

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

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

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

{ $\Delta = 0$;

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

{

$W(s) = \sum_{a_j} \pi(s,a_j) \sum_{s'} P^{a_j}_{s->s'} \left[R^{a_j}_{s->s'} + \gamma V(s') \right]$ // W(s) è $V_{k+1}(s)$

$\Delta = \max(\Delta, | V(s) - W(s) |)$

}

for s=1:N

$V(s) = W(s)$;

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

Interpretazione dell'update (batch o trial)

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

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

$$V_{k+1}(s) = \sum_{a_j} \pi(s,a_j) \sum_{s'} P^{a_j}_{s->s'} \left[R^{a_j}_{s->s'} + \gamma V(s') \right]$$

Forward pass

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

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

3 assunzioni:

1) Conoscenza della dinamica dell'ambiente: P(s->s'| a_j)
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 $\pi(.)$, se questa si basa sul reward che riceverò negli istanti futuri?

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

Sommario

Determinazione ricorsiva della Value function

Determinazione della policy ottima.

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).

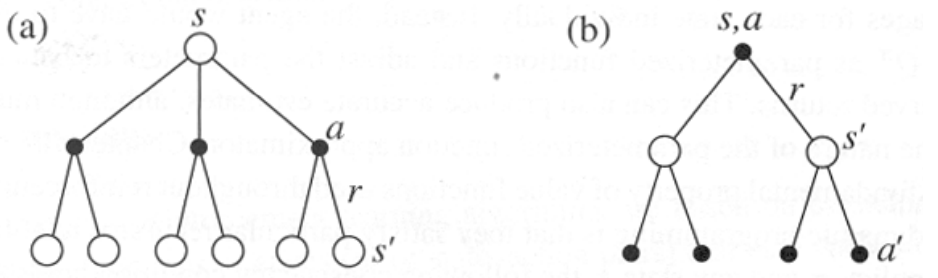
Per tutte le possibili azioni devo calcolare la probabilità di transizione allo stato successivo e di ottenere una certa reward.

3 assunzioni:

1) Conoscenza della dinamica dell'ambiente: P(s->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.

Relazione soddisfatta da V*(s)

$$V^*(s) = \underset{a}{Max}\left[E_\pi\{R_t \mid s_t = s, a_t\}\right] =$$

$$\underset{a}{Max}\left[E_\pi\left\{\sum_{k=0}^{\infty}\gamma^k r_{t+k+1} \,\middle|\, s_t = s, a_t = a\right\}\right] =$$

$$\underset{a}{Max}\left[r_{t+1} + \gamma E_\pi\left\{\sum_{k=0}^{\infty}\gamma^k r_{t+k+2} \,\middle|\, s_t = s, a_t = a\right\}\right] =$$

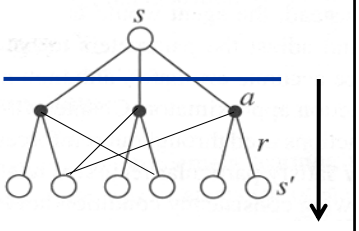
$$\underset{a}{Max}\left[r_{t+1} + \gamma V^*(s_{t+1}) \,|\, s_t = s, a_t = a\right] =>$$

$$V^*(s) = \underset{a}{Max}\left\{P_{s->s'|a}\left[R_{s->s'|a} + \gamma V^*(s')\right]\right\}$$

Bellman's Equation For optimal policy

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?

Miglioramento della policy

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

Condizioni di funzionamento dell'agente:
Policy **deterministica**: a = π(s).
Ambiente **stocastico**.

Cosa succede se cambiamo la policy per un certo stato s? a' ≠ $\pi(s_m)$.
Cosa viene influenzato?

Scelgo a' in s, visiterò una certa sequenza di stati, per questi stati seguirò la policy precedente per s ≠ sm.
Cosa viene influenzato?

Come faccio a valutare se miglioro la policy o no?

Effetto del cambiamento della policy

Cambia, a, cambiano i possibili stati successivi ad s, $\{s_{t+1}\}$, ed il reward a lungo termine:

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

$$\sum_{s'} P^{a_{new}}_{s_m \to s'}\left[R^{a_{new}}_{s_m \to s'} + \gamma V^{\pi}(s')\right]$$

?

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

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

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).

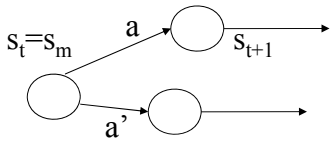
Enunciato del teorema del miglioramento della policy

$$Q^{\pi}(s,a)=\sum_{k} P_{s\to s_k|a}\left[R_{s\to s_k|a}+\gamma V^{\pi}(s_k)\right]$$

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

$$Q^{\pi'}(s,a_{new}=\pi'(s_m,a))=\sum_{k} P_{s_m\to s_k|a_{new}}\left[R_{s_m\to s_k|a_{new}}+\gamma V^{\pi}(s_k)\right]$$

Tesi: π' è meglio di π. Cioè: $V^{\pi'}(s) >= V^{\pi}(s)$ ∀s.



Dimostrazione del teorema del miglioramento della policy

Analizziamo la seguente condizione:

π'= π ∀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 $a_{new} = \pi'(s)$.

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

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

Dimostrazione del teorema del miglioramento della policy

Hp: $Q^{\pi}(s,\pi'(s)) >= V^{\pi}(s)$ $\forall s$ $\pi'(s,a)$ è migliore per almeno uno stato

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

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

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

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

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

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

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

Th: $V^{\pi}(s) <= V^{\pi'}(s)$

Ottimizzazione policy

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

Posso ordinare la Value function V(s) in funzione delle azioni scelte in s (policy).

Si definisce una policy, π_1, migliore di un'altra, π_2, se e solo se:
$V^{\pi 1}(s) > = V^{\pi 2}(s)$ $\forall s$.

In particolare si definisce una politica ottima, π^*, se e solo se:
$V^*(s) > = V^{\pi}(s)$ $\forall s$

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

Calcolo ricorsivo della Value function ottima::confronti

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

V*(s) di uno stato, quando viene scelta la policy ottima,
deve essere uguale al valore atteso del reward per l'azione
migliore per lo stato s.

$$V^{*}(s)=\max_{a_j}\sum_{s'}P_{s\to s'|a_j}\left[R_{s\to s'|a_j}+\gamma V^{*}(s')\right]$$

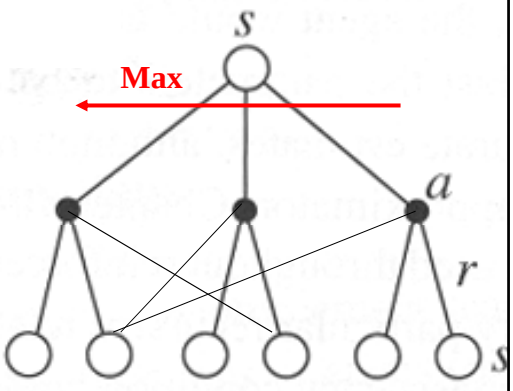
Politica greedy: scelgo l'azione ottimale.
Ha senso per il robot raccogli-lattine?

V*(s) – Osservazioni

$$V^{*}(s)=\max_{a_j}\sum_{s'}P_{s\to s'|a_j}\left[R_{s\to s'|a_j}+\gamma V^{*}(s')\right]=\max_{a_j}Q(s,a_j)$$

Per ogni stato devo valutare:
•L'azione migliore ad un passo

Come valuto?
• analizzando reward a lungo termine



t

Utilizzo di V(s) per determinare la policy ottima

Esplorazione dell'effetto a lungo termine di tutte le azioni possibili.

Politica greedy rispetto alla Value function.

Questa politica greedy (ad un passo) produce una politica ottima globalmente.

Vengono valutate le conseguente a breve termine delle azioni (1-step) ma non è una politca miope perché consente di ottenere una politica globalmente ottima.

E' reso possibile per la conoscenza dell'ambiente (stocastico).

Politica ottima

Miglioramento della politica per tutti gli stati.

$$\pi'(s_k) = \arg\max_a Q^\pi(s,a) \qquad \textit{greedy o } \varepsilon\textit{-greedy}$$

$$\forall s \quad = \arg\max_a E\{r_{t+1} + \gamma V^\pi(s') \mid s_t = s, a_t = a\}$$

$$= \arg\max_a \sum_{s'} P^a_{s->s'}\left[R^a_{s->s'} + \gamma V^\pi(s')\right]$$

Policy improvement

Si può estendere al caso di comportamento stocastico dell'agente nel qual caso: $\pi(s,a)$ è una probabilità. Questo sarà utile quando l'ambiente non sarà più considerato noto.

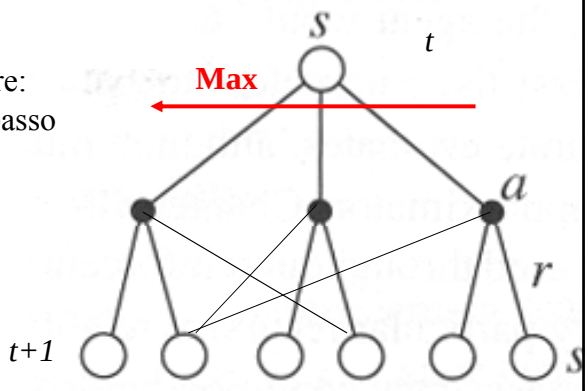
Policy ottima

$$V^*(s) = \max_{a_j} \sum_{s'} P_{s \to s' | a_j} \left[R_{s \to s' | a_j} + \gamma V^*(s') \right]$$

$$\pi^*(s) = \arg\max_a Q^\pi(s,a) \qquad \forall s$$

Per ogni stato devo valutare:
•L'azione migliore ad un passo

Come valuto?
• analizzando reward a lungo termine



Policy iteration

Iterazione tra:

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

$$\pi_0 \rightarrow V^{\pi 0} \rightarrow \pi_1 \rightarrow V^{\pi 1} \rightarrow \pi_2 \rightarrow V^{\pi 2} \rightarrow \ldots\ldots$$
$$\rightarrow \pi^* \rightarrow V^*$$

Converge velocemente ad una buona politica

Algoritmo (progetto per esame)

Repeat until policy-stable

1. Initialization
   $V(s) \in \Re$ and $\pi(s) \in \mathcal{A}(s)$ arbitrarily for all $s \in \mathcal{S}$

2. Policy Evaluation
   Repeat
   $\Delta \leftarrow 0$
   For each $s \in \mathcal{S}$:
   $v \leftarrow V(s)$
   $V(s) \leftarrow \sum_{s'} \mathcal{P}^{\pi(s)}_{ss'} \left[\mathcal{R}^{\pi(s)}_{ss'} + \gamma V(s') \right]$
   $\Delta \leftarrow \max(\Delta, |v - V(s)|)$
   until $\Delta < \theta$ (a small positive number)

3. Policy Improvement
   policy-stable $\leftarrow$ true
   For each $s \in \mathcal{S}$:
   $b \leftarrow \pi(s)$
   $\pi(s) \leftarrow \arg\max_a \sum_{s'} \mathcal{P}^a_{ss'} \left[\mathcal{R}^a_{ss'} + \gamma V(s') \right]$
   If $b \neq \pi(s)$, then *policy-stable* $\leftarrow$ false
   If *policy-stable*, then stop; else go to 2

Politica ottima

Miglioramento della politica per tutti gli stati.

$$\pi'(s_k) = \arg\max_a Q^\pi(s,a)$$

$$= \arg\max_a Q^\pi(s,a)$$

$\forall s$

$$= \arg\max_a E\{r_{t+1} + \gamma V^\pi(s') \mid s_t = s, a_t = a\}$$

$$= \arg\max_a \sum_{s'} P^a_{s->s'} \left[R^a_{s->s'} + \gamma V^\pi(s') \right]$$

Si può estendere al caso di comportamento stocastico dell'agente nel qual caso: $\pi(s,a)$ è una probabilità.

Policy iteration

Iterazione tra:

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

$$\pi_0 \rightarrow V^{\pi 0} \rightarrow \pi_1 \rightarrow V^{\pi 1} \rightarrow \pi_2 \rightarrow V^{\pi 2} \rightarrow \ldots\ldots\ldots \rightarrow \pi^* \rightarrow V^*$$

Converge velocemente ad una buona politica

Iterative policy evaluation - problema

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

Converge al limite a $V^{\pi}(s)$. Come facciamo a troncare?

Value iteration

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

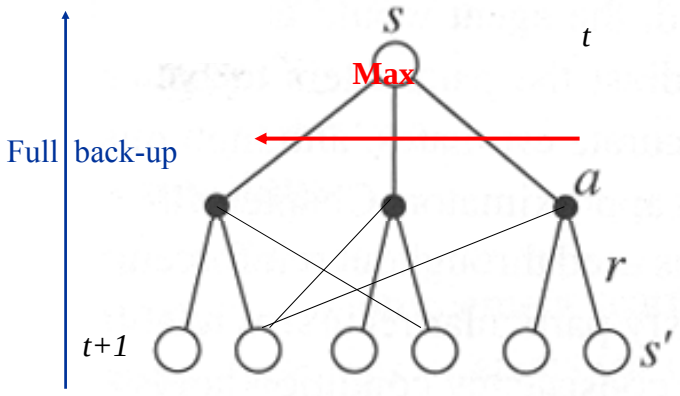
Invece di considerare una policy stocastica, consideriamo l'azione migliore:

$$V_{k+1}(s) = \max_a \sum_{s'} P_{s \to s'|a} \left[R_{s \to s'|a} + \gamma V_k(s')\right]$$

$\forall s$

Visualizzazione grafica

$$V_{k+1}(s) = \max_a \sum_{s'} P_{s \to s'|a} \left[R_{s \to s'|a} + \gamma V_k(s')\right]$$



Confronto con l'equazione di Bellman

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

V*(s) di uno stato, quando viene scelta la policy ottima, deve essere uguale al valore atteso del reward per l'azione migliore per lo stato s.

$$V^{*}(s)=\max_{a_j}\sum_{s'}P_{s\to s'|a_j}\left[R_{s\to s'|a_j}+\gamma V^{*}(s')\right]$$

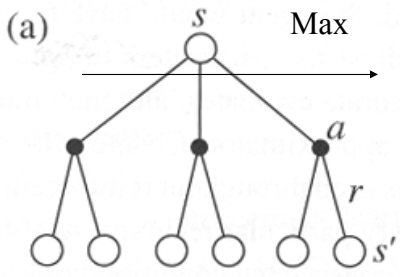
$$V_{k+1}(s)=\max_{a_j}\sum_{s'}P_{s\to s'|a}\left[R_{s\to s'|a_j}+\gamma V_k(s')\right]$$

Confronto con l'equazione di backup

$$V_{k+1}(s)=\max_{a}\sum_{s'}P_{s\to s'|a}\left[R_{s\to s'|a}+\gamma V_k(s')\right]$$

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

Nel caso della Value Iteration viene considerata solamente l'azione che fornisce il valore massimo.



Algoritmo di value iteration

V(s) = 0 ∀s compreso TS

Repeat
{ Δ = 0;
for s = 1:NS // Per ogni stato eccetto il TS
{ V_buffer = V(s);

$$V(s) = \max_a \left\{ \sum_{s'} P_{s->s'|a} \left[R_{s->s'|a} + \gamma V(s') \right] \right\}$$

Δ = max(Δ, |V(s) – V_buffer|);
}
} until (Δ < Th);

Output a deterministic policy such that:

$$\pi(s) = \arg\max_a \sum_{s'} P_{s\to s'|a} \left[R_{s\to s'|a} + \gamma V(s') \right]$$

Sommario

Determinazione ricorsiva della Value function

Determinazione della policy ottima.