UNIVERSITÀ DEGLI STUDI DI SALERNO
DIPARTIMENTO DI INFORMATICA

Intelligenza Artificiale

Processi Decisionali di Markov

a.a. 2022/2023

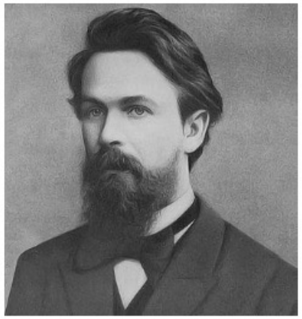
Outline

- Processi di Markov
- Processi di ricompensa di Markov
- Processi decisionali di Markov
- Estensioni del MDP

Introduzione ai MDP

- I processi decisionali di Markov descrivono formalmente un ambiente per il RL
  - Quando l'ambiente è completamente osservabile
  - Cioè, lo stato attuale caratterizza completamente il processo
- Quasi tutti i problemi di RL possono essere formalizzati come MDP, ad esempio
  - I problemi parzialmente osservabili possono essere convertiti in MDP
  - Il controllo ottimale si occupa principalmente di MDP continui

Proprietà di Markov



Dato il presente, il futuro è indipendente dal passato

- Uno stato S_t è detto di Markov se e solo se

$$P[S_{t+1} \mid S_t] = P[S_{t+1} \mid S_1, \ldots, S_t]$$

- Lo stato raccoglie tutte le informazioni rilevanti della storia
- Una volta che lo stato è noto, la storia può non essere considerata
- Lo stato è una statistica sufficiente del futuro

Matrice di transizione di stato

- Per uno stato di Markov s e uno stato successore s', la *probabilità di transizione di stato* è definita da

$$\mathcal{P}_{ss'} = \mathbb{P}\left[S_{t+1} = s' \mid S_t = s\right]$$

- La matrice di transizione di stato P definisce le probabilità di transizione da tutti gli stati s a tutti gli stati successori s',

$$\mathcal{P} = \textit{from} \begin{array}{c} \textit{to} \\ \begin{bmatrix} \mathcal{P}_{11} & \dots & \mathcal{P}_{1n} \\ \vdots & & \\ \mathcal{P}_{n1} & \dots & \mathcal{P}_{nn} \end{bmatrix} \end{array}$$

- La somma dei valori in ogni riga è pari a 1

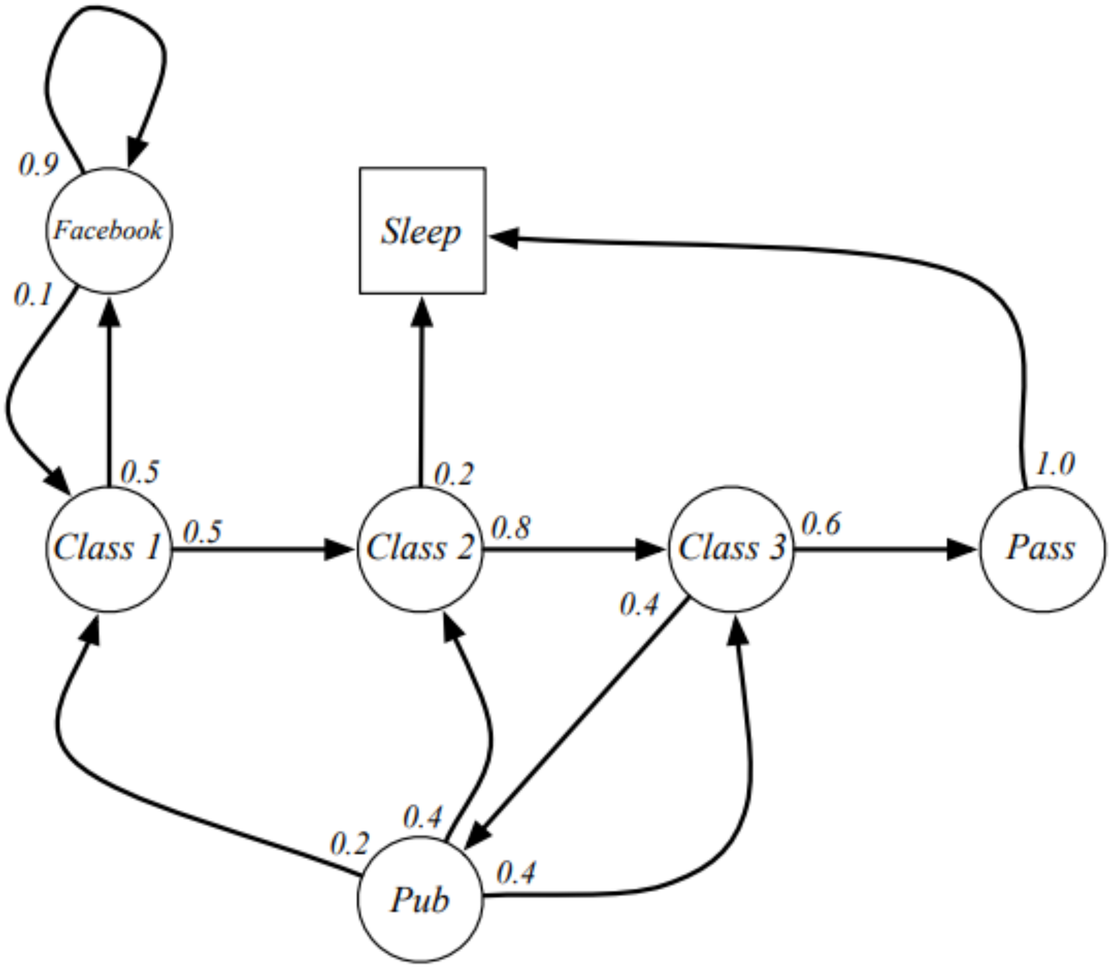
Processo di Markov

- Un processo di Markov è un processo casuale senza memoria, cioè una sequenza di stati casuali $S_1, S_2, \ldots$ caratterizzati dalla proprietà di Markov

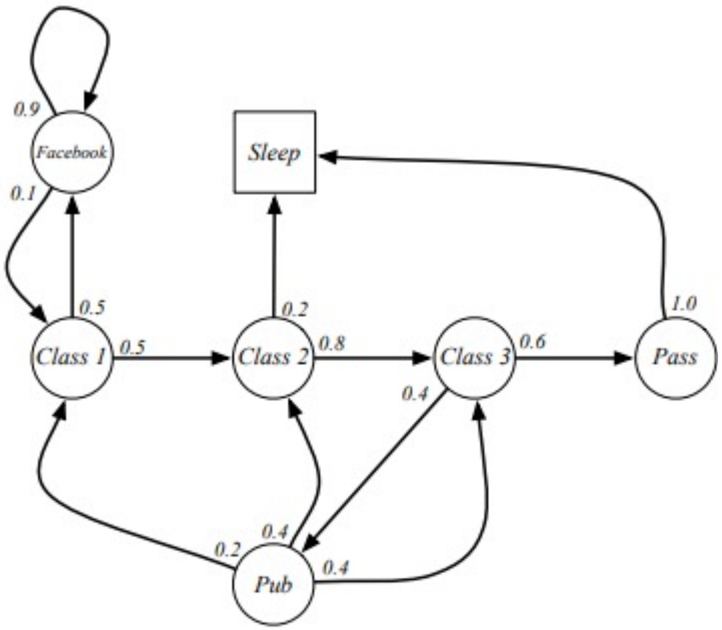
- Un *processo di Markov* (o *catena di Markov*) è una tupla *<S, P>*
  - S è un insieme (finito) di stati
  - P è una matrice di probabilità di transizione di stato

$$P_{ss'} = \mathrm{P}[S_{t+1} = s' \mid S_t = \mathrm{s}]$$

Esempio: Catena di Markov degli studenti



Esempio: Episodi della Catena di Markov degli studenti

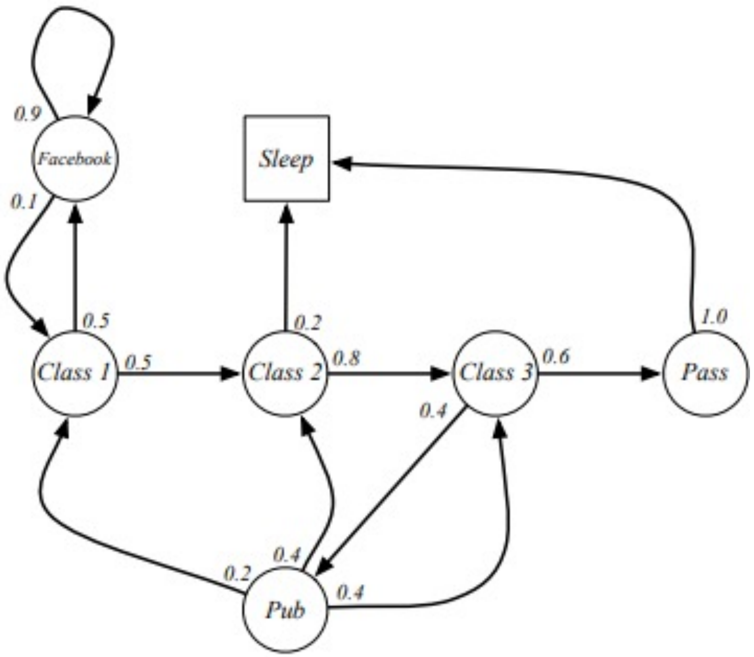


- Esempi di episodi ottenuti partendo da S_1=*Class1* (C_1)

$$S_1, S_2, \ldots, S_t$$

- C_1 C_2 C_3 Pass Sleep
- C_1FB FB C_1 C_2 Sleep
- C_1 C_2 C_3 Pub C_2 C_3 Pass Sleep
- C_1FB FB C_1 C_2 C_3 Pub C_1FB FB FB C_1 C_2 C_3 Pub C_2 Sleep

Esempio: Matrice di transizione della Catena di Markov degli studenti



$$P_{ss'} = P(S_{t+1} = s' \mid S_t = s)$$

Future

$$\mathcal{P} = \begin{array}{cc} & \begin{array}{ccccccc} C1 & C2 & C3 & Pass & Pub & FB & Sleep \end{array} \\ \begin{array}{c} C1 \\ C2 \\ C3 \\ Pass \\ Pub \\ FB \\ Sleep \end{array} & \begin{bmatrix} & 0.5 & & & & 0.5 & \\ & & 0.8 & & & & 0.2 \\ & & & 0.6 & 0.4 & & \\ & & & & & & 1.0 \\ 0.2 & 0.4 & 0.4 & & & & \\ 0.1 & & & & & 0.9 & \\ & & & & & & 1 \end{bmatrix} \end{array}$$

Current

Processo di ricompensa di Markov

- Un processo di ricompensa di Markov (MRP) è una catena di Markov con valori

- Un *processo di ricompensa di Markov* è una tupla <*S*, *P*, *R*, *γ*>
  - S è un insieme (finito) di stati
  - P è una matrice di probabilità di transizione di stato

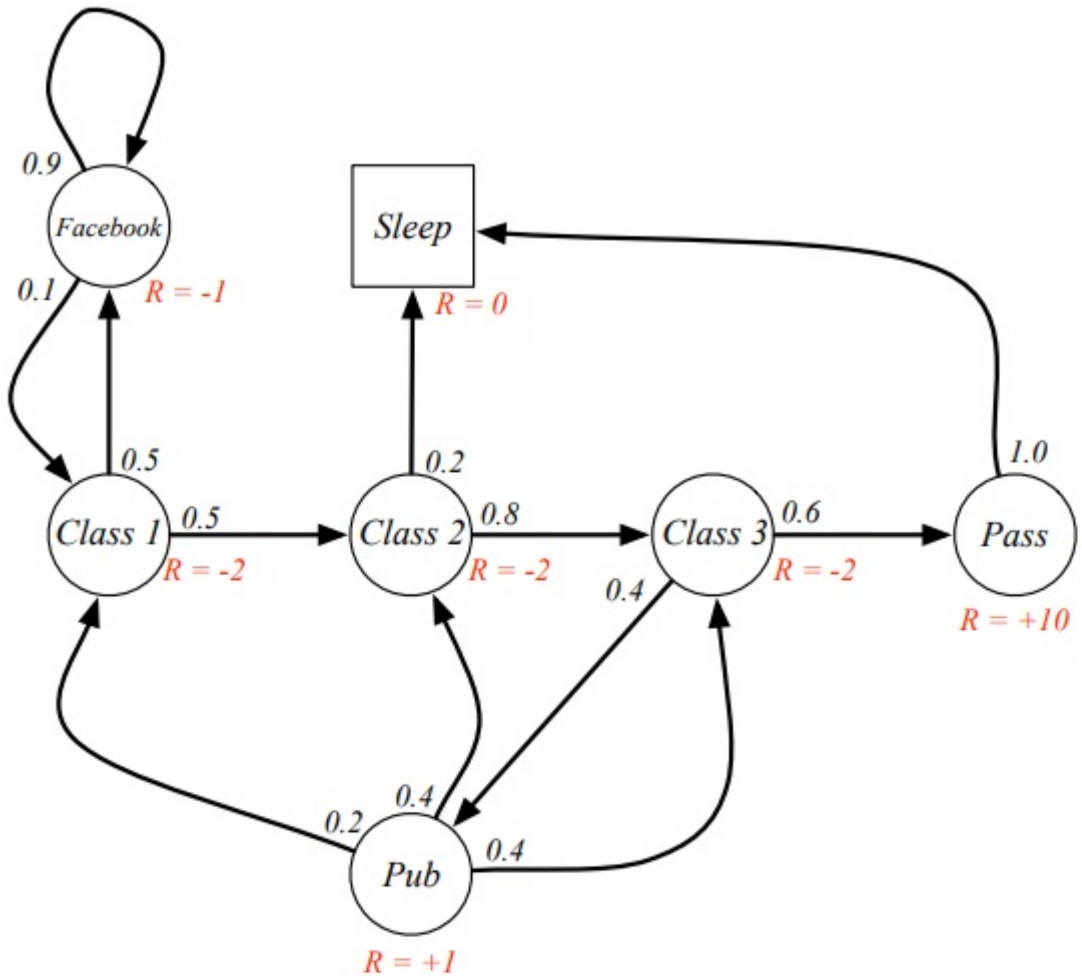
$$P_{ss'} = \mathrm{P}[S_{t+1} = s' \mid S_t = \mathrm{s}]$$

  - R è una funzione di ricompensa,

$$R_s = \mathrm{E}[R_{t+1} \mid S_t = \mathrm{s}]$$

  - γ è un fattore di sconto, $\gamma \in [0,1]$

Esempio: MRP degli studenti



Guadagno

- Il *guadagno* G_t è la ricompensa totale calcolata a partire dal time-step *t*

$$G_t = R_{t+1} + \gamma R_{t+2} + \ldots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

- Lo sconto $\gamma \in [0,1]$ è il valore attuale delle ricompense future
- Il valore della ricompensa ricevuta *R* dopo *k+1* time-step è $\gamma^k R$
- In questo modo la ricompensa immediata viene privilegiata rispetto a quella a lungo termine
  - γ vicino a 0 porta a una valutazione "miope"
  - γ vicino a 1 porta a una valutazione "lungimirante"

Perché lo sconto?

- La maggior parte dei processi decisionali e di ricompensa di Markov sono «*scontati*». Perché?
  - Matematicamente conveniente scontare i premi
  - Evita guadagni infiniti nei processi di Markov ciclici
  - L'incertezza sul futuro potrebbe non essere pienamente rappresentata
    - Se la ricompensa è di tipo finanziaria, le ricompense immediate possono fruttare più interessi di quelle a lungo termine
    - Il comportamento animale/umano mostra una preferenza per la ricompensa immediata
  - A volte è possibile utilizzare processi di ricompensa di Markov non scontati (cioè $\gamma = 1$), ad esempio se tutte le sequenze terminano

Value function

- La value function misura il valore a lungo termine di essere in uno stato s

- La *state-value function v(s)* di un MRP è il guadagno previsto partendo dallo stato s

$$v(s) = \mathrm{E}[G_t \mid S_t = s]$$

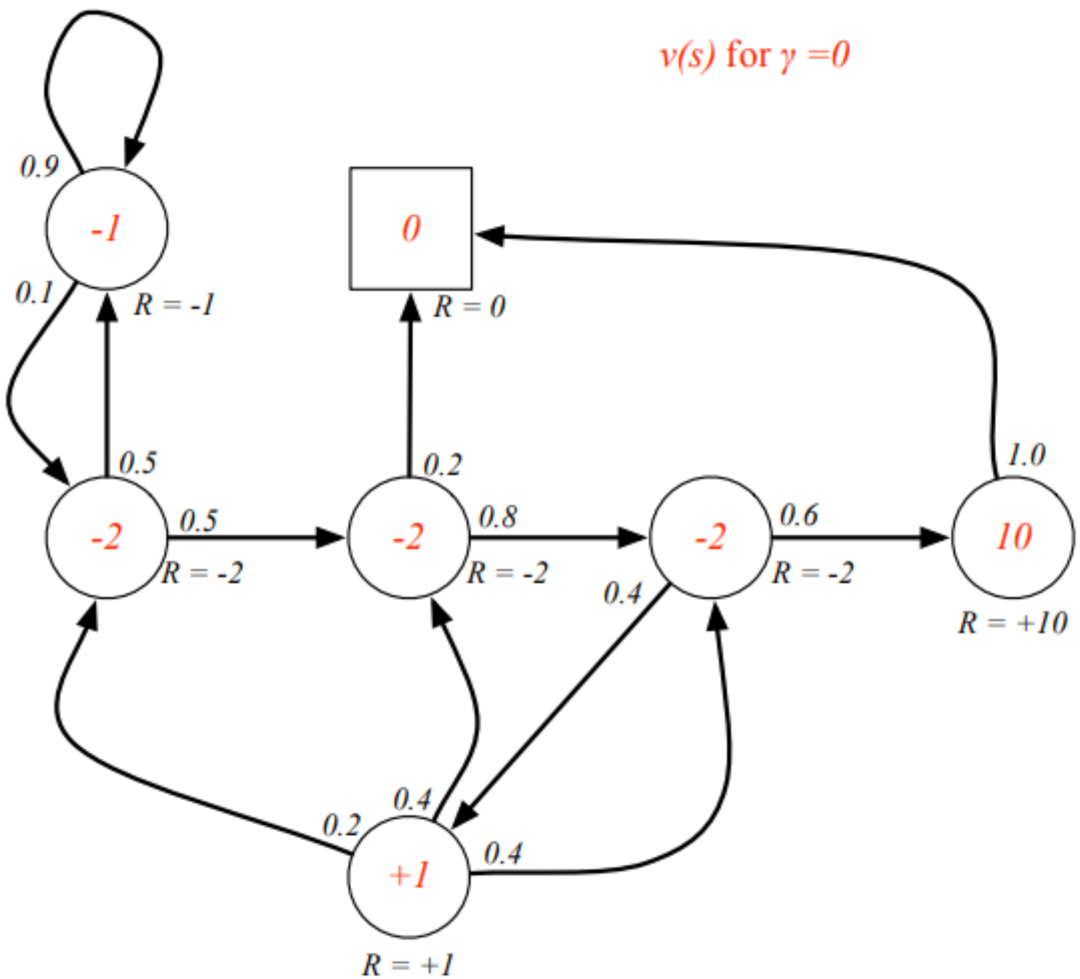
Esempio: Guadagni nel MRP degli studenti

- Esempio dei guadagni ottenuti partendo da $S_1 = C_1$ e con $\gamma = \frac{1}{2}$

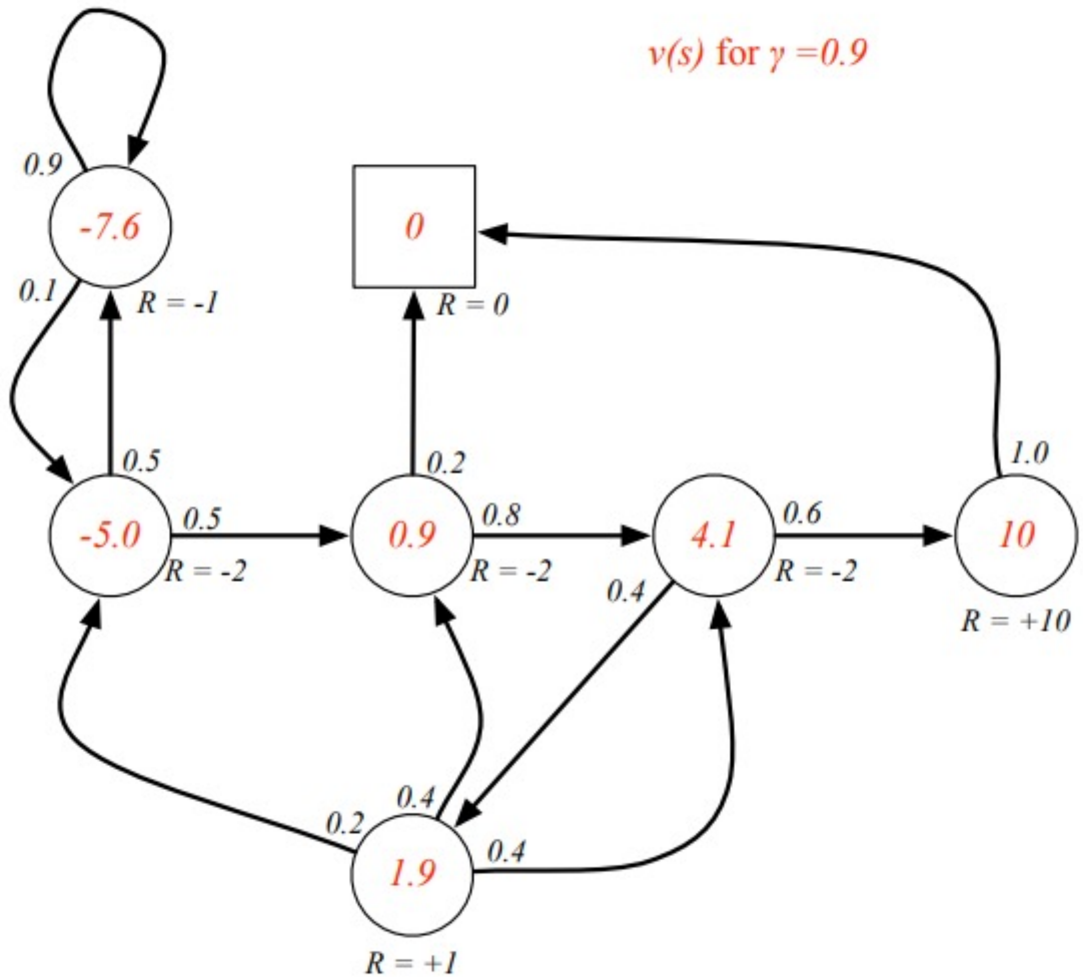
$$G_1 = R_2 + \gamma R_3 + \ldots + \gamma^{T-2} R_T$$

C1 C2 C3 Pass Sleep	$v_1 = -2 - 2 * \frac{1}{2} - 2 * \frac{1}{4} + 10 * \frac{1}{8}$	$=$	-2.25
C1 FB FB C1 C2 Sleep	$v_1 = -2 - 1 * \frac{1}{2} - 1 * \frac{1}{4} - 2 * \frac{1}{8} - 2 * \frac{1}{16}$	$=$	-3.125
C1 C2 C3 Pub C2 C3 Pass Sleep	$v_1 = -2 - 2 * \frac{1}{2} - 2 * \frac{1}{4} + 1 * \frac{1}{8} - 2 * \frac{1}{16} \ldots$	$=$	-3.41
C1 FB FB C1 C2 C3 Pub C1 ...	$v_1 = -2 - 1 * \frac{1}{2} - 1 * \frac{1}{4} - 2 * \frac{1}{8} - 2 * \frac{1}{16} \ldots$	$=$	-3.20
FB FB FB C1 C2 C3 Pub C2 Sleep			

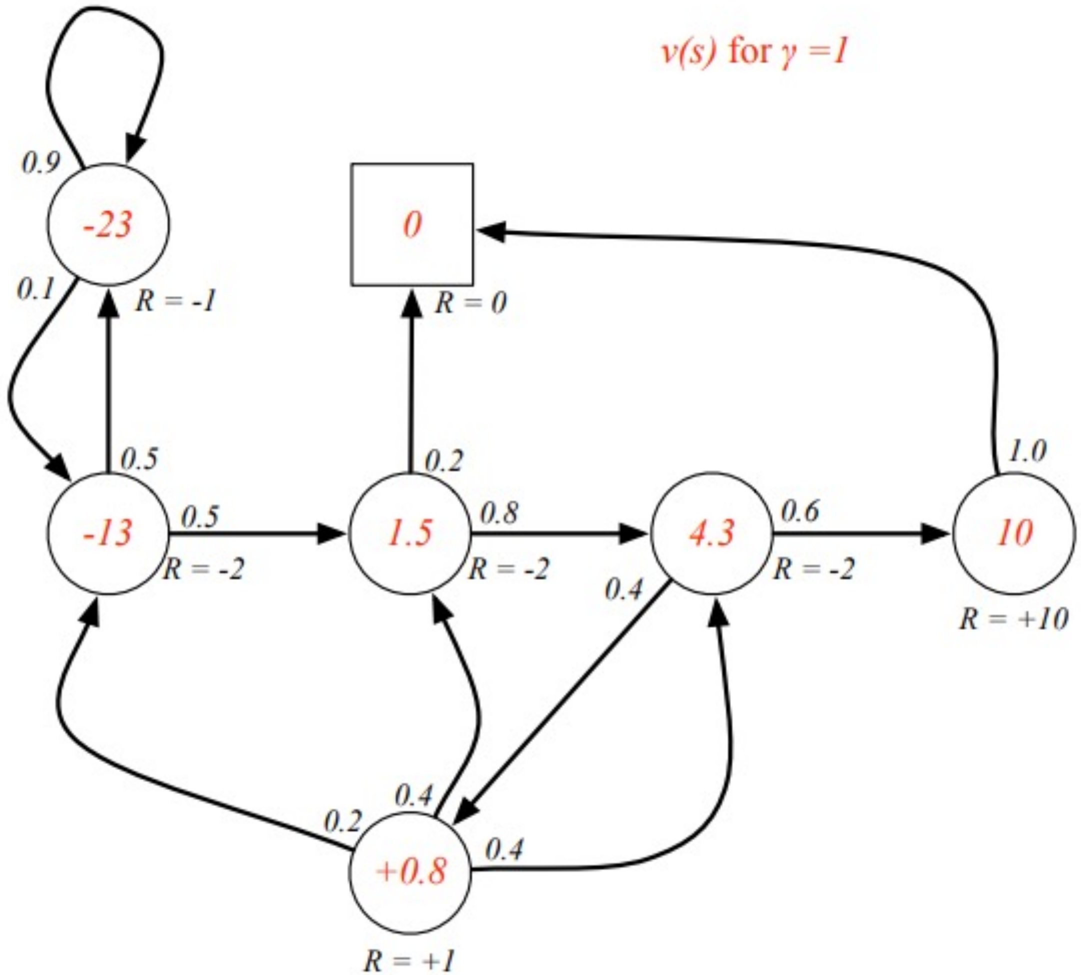
Esempio: State-Value Function per il MRP degli studenti



Esempio: State-Value Function per il MRP degli studenti (2)



Esempio: State-Value Function per il MRP degli studenti (3)



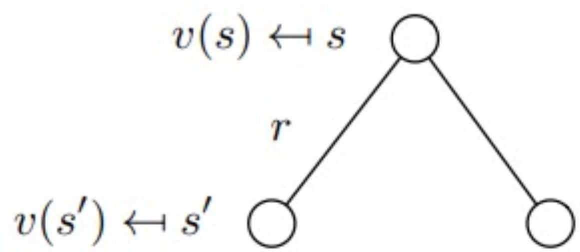
Bellman Equation per MRP

- La value function può essere suddivisa in due parti:
  - ricompensa immediata R_{t+1}
  - valore scontato dello stato successore $\gamma\, v(S_{t+1})$

$$
\begin{aligned}
v(s) &= \mathbb{E}\left[G_t \mid S_t = s\right] \\
&= \mathbb{E}\left[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots \mid S_t = s\right] \\
&= \mathbb{E}\left[R_{t+1} + \gamma \left(R_{t+2} + \gamma R_{t+3} + \ldots\right) \mid S_t = s\right] \\
&= \mathbb{E}\left[R_{t+1} + \gamma G_{t+1} \mid S_t = s\right] \\
&= \mathbb{E}\left[R_{t+1} + \gamma v(S_{t+1}) \mid S_t = s\right]
\end{aligned}
$$

Bellman Equation per MRP (2)

$$v(s) = \mathbb{E}\left[R_{t+1} + \gamma v(S_{t+1}) \mid S_t = s\right]$$

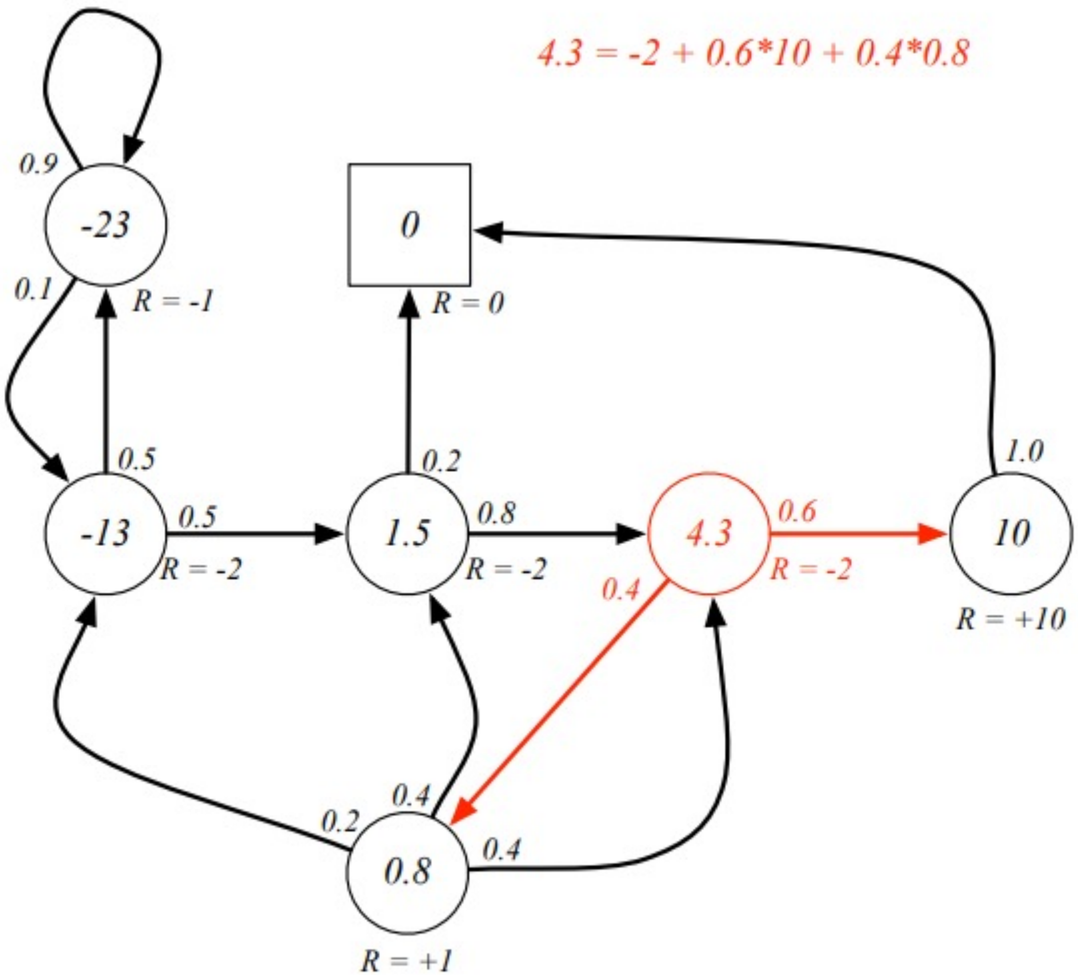


$$v(s) = \overbrace{\mathbb{E}[R_{t+1} | S_t = s]}^{\text{Reward function } \mathcal{R}_s} + \gamma \mathbb{E}[v(S_{t+1}) | S_t = s]$$

The expected state-value of being in any state reachable from *s*

$$v(s) = \mathcal{R}_s + \gamma \sum_{s'} P_{ss'} v(s')$$

Esempio: Bellman Equation per il MRP degli studenti



Bellman Equation in forma matriciale

- La Bellman Equation può essere espressa in modo conciso utilizzando le matrici

$$v = \mathcal{R} + \gamma \mathcal{P} v$$

- dove v è un vettore colonna con una entry per ogni stato

$$\begin{bmatrix} v(1) \\ \vdots \\ v(n) \end{bmatrix} = \begin{bmatrix} \mathcal{R}_1 \\ \vdots \\ \mathcal{R}_n \end{bmatrix} + \gamma \begin{bmatrix} \mathcal{P}_{11} & \dots & \mathcal{P}_{1n} \\ \vdots & & \\ \mathcal{P}_{11} & \dots & \mathcal{P}_{nn} \end{bmatrix} \begin{bmatrix} v(1) \\ \vdots \\ v(n) \end{bmatrix}$$

Risoluzione Bellman Equation

- La Bellman Equation è un'equazione lineare
- Essa può essere risolta direttamente:

$$
\begin{aligned}
v &= \mathcal{R} + \gamma \mathcal{P} v \\
(I - \gamma \mathcal{P})\, v &= \mathcal{R} \\
v &= (I - \gamma \mathcal{P})^{-1} \mathcal{R}
\end{aligned}
$$

- La complessità computazionale è O(n^3) per *n* stati
- È possibile applicare la soluzione diretta solo per MRP di piccole dimensioni
- Esistono molti metodi iterativi per i MRP di grandi dimensioni, ad es.
  - Programmazione dinamica
  - Temporal-Difference Learning
  - Valutazione di Monte-Carlo

Processo decisionale di Markov

- Un processo decisionale di Markov (MDP) è un processo di ricompensa di Markov con decisioni
- L'*ambiente* è costituito da stati di Markov
- Un *processo decisionale di Markov* è una tupla <*S*, *A*, *P*, *R*, *γ*>
  - S è un insieme finito di stati
  - A è un insieme finito di azioni
  - P è una matrice di probabilità di transizione di stato

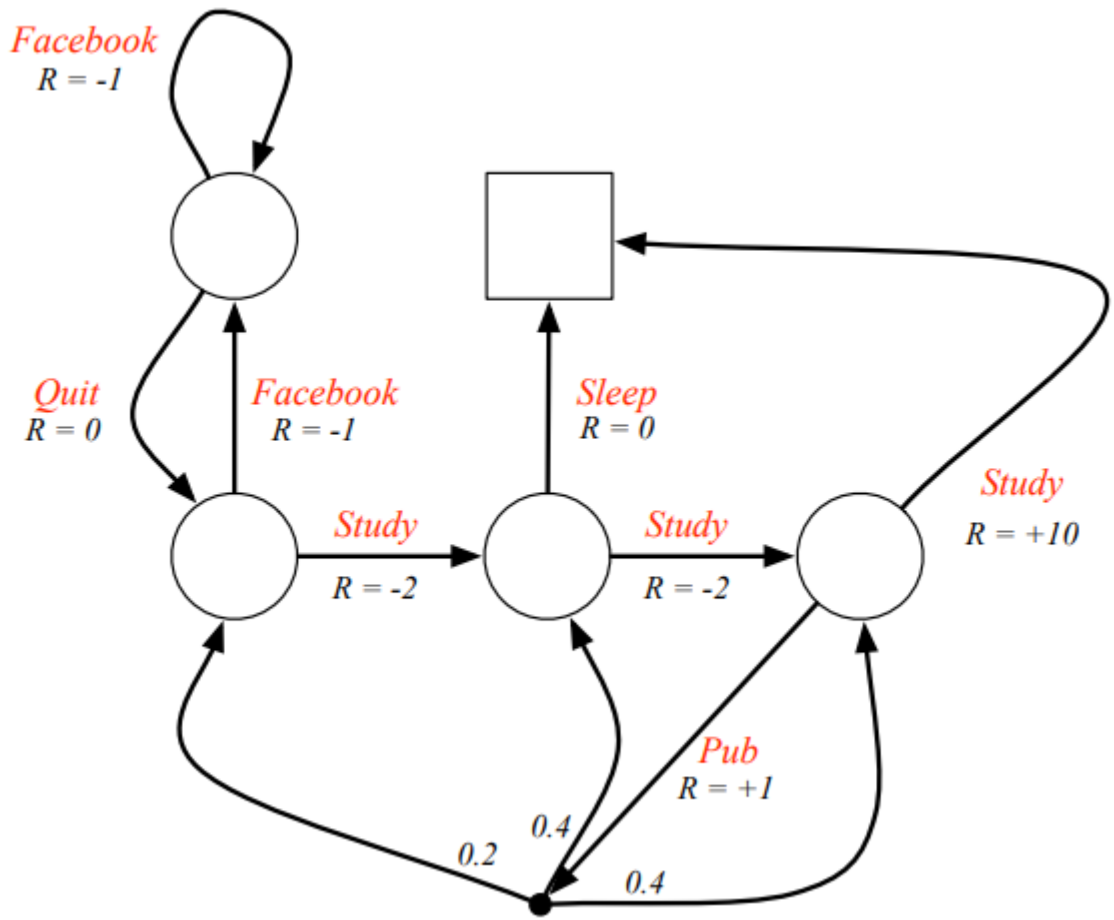
$$P_{ss'}^{a} = \mathrm{P}[S_{t+1} = s' \mid S_t = s, A_t = a]$$

  - R è una funzione di ricompensa,

$$R_s^a = \mathrm{E}[R_{t+1} \mid S_t = s, A_t = a]$$

  - γ è un fattore di sconto, $\gamma \in [0,1]$

Esempio: MDP degli studenti



Policy

- Una *policy* π è una distribuzione sulle azioni in funzione degli stati

$$\pi(a|s) = P\,[A_t = a \mid S_t = s]$$

- Una policy definisce in modo completo il comportamento di un agente
- Le policy di un MDP dipendono solo dallo stato attuale (non dalla storia)
- Le policy sono stazionarie (indipendenti dal tempo)

$$A_t \sim \pi(\cdot \mid S_t), \forall t > 0$$

Policy (2)

- Dato un MDP <*S, A, P, R, γ*> e una policy π
- La sequenza di stati $S_1, S_2, \ldots$ è un processo di Markov <*S*, P^{π} >
- La sequenza di stati e ricompense $S_1, R_1, S_2, \ldots$ è un processo di ricompensa di Markov <*S*, P^{π} , R^{π} , γ> dove

$$\mathcal{P}^{\pi}_{s,s'} = \sum_{a \in \mathcal{A}} \pi(a|s) \mathcal{P}^{a}_{ss'}$$

$$\mathcal{R}^{\pi}_{s} = \sum_{a \in \mathcal{A}} \pi(a|s) \mathcal{R}^{a}_{s}$$

Value function

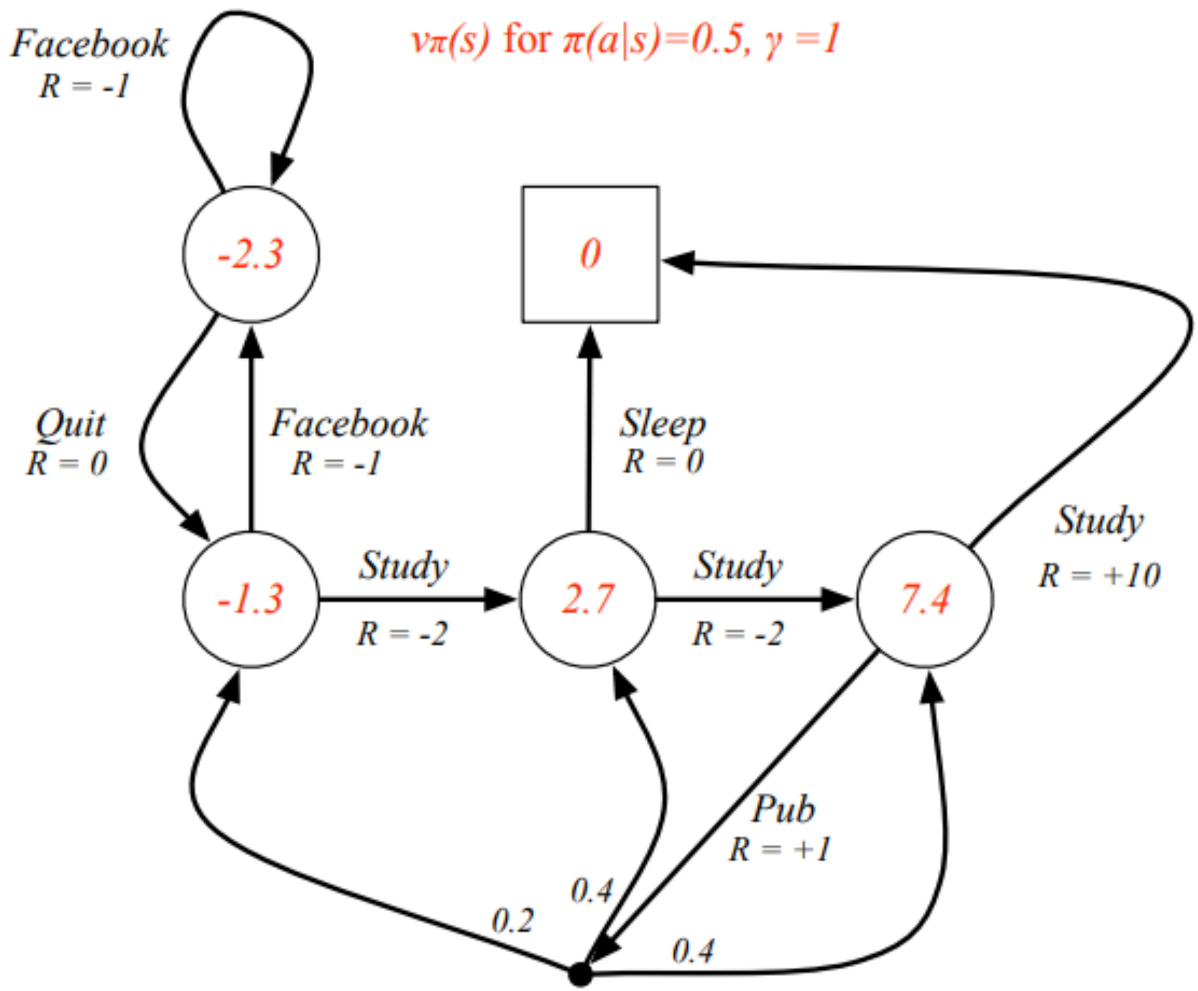
- La *state-value function* $v_\pi(s)$ di un MDP è il guadagno atteso partendo dallo stato *s* e seguendo la policy π

$$v_\pi(s) = E_\pi[G_t \mid S_t = \mathrm{s}]$$

- La *action-value function* $q_\pi(s, a)$ è il guadagno atteso partendo dallo stato *s*, eseguendo l'azione *a* e seguendo la policy π

$$q_\pi(s, a) = E_\pi[G_t \mid S_t = \mathrm{s}, A_t = \mathrm{a}]$$

Esempio: State-Value Function per il MDP degli studenti



Bellman Expectation Equation

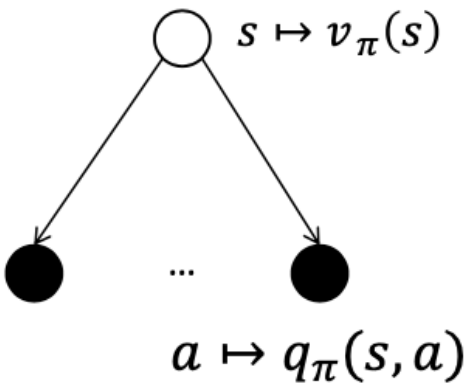
- La state-value function può essere nuovamente scomposta in ricompensa immediata più valore scontato dello stato successivo,

$$v_\pi(s) = \mathbb{E}_\pi\left[R_{t+1} + \gamma v_\pi(S_{t+1}) \mid S_t = s\right]$$

- La action-value function può essere scomposta in modo analogo,

$$q_\pi(s,a) = \mathbb{E}_\pi\left[R_{t+1} + \gamma q_\pi(S_{t+1}, A_{t+1}) \mid S_t = s, A_t = a\right]$$

Bellman Expectation Equation per v_π

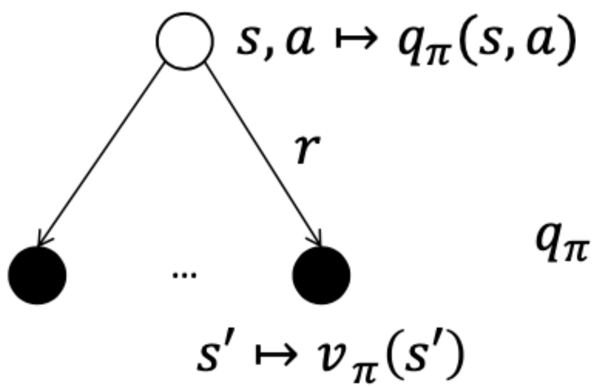


$$v_\pi(s) = \mathbb{E}_\pi[R_{t+1} + \gamma v_\pi(S_{t+1}) | S_t = s]$$

Expectation with respect to the actions that can be taken starting from *s*

$$v_\pi(s) = \sum_{a \in \mathcal{A}} \pi(a|s) q_\pi(s,a)$$

Bellman Expectation Equation per q_π

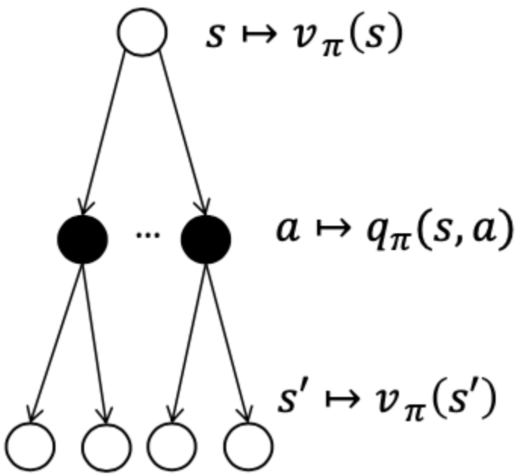


$$q_\pi(s,a) = \mathbb{E}_\pi[R_{t+1} + \gamma q_\pi(S_{t+1}, A_{t+1}) | S_t = s, A_t = a]$$

Expectation with respect to the states reachable from s having taken action a

$$q_\pi(s,a) = \mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} P_{ss'}^a v_\pi(s')$$

Bellman Expectation Equation per v_π(2)

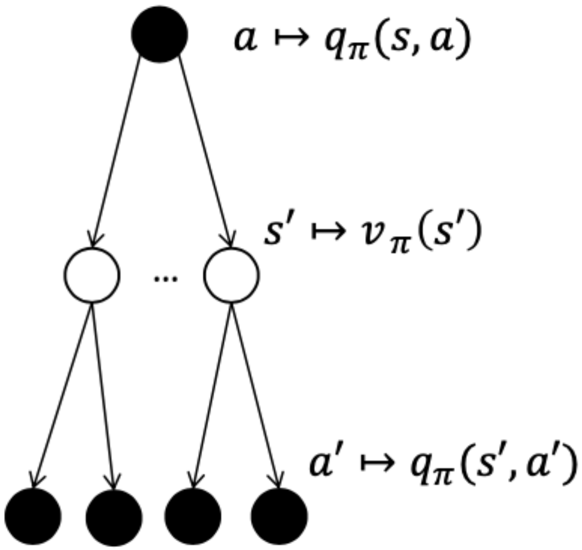


$$v_\pi(s) = \sum_{a \in \mathcal{A}} \pi(a|s) q_\pi(s,a)$$

The expected return of being in a state reachable from s through action a and then continue following policy

$$v_\pi(s) = \sum_{a \in \mathcal{A}} \pi(a|s) \left(\mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} P_{ss'}^a v_\pi(s') \right)$$

Bellman Expectation Equation per q_π(2)



$$q_\pi(s,a) = \mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} P_{ss'}^a v_\pi(s')$$

The expected return of any action a' taken from states reachable from s through action a (and then follow policy)

$$q_\pi(s,a) = \mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} P_{ss'}^a \sum_{a' \in \mathcal{A}} \pi(a'|s') q_\pi(s',a')$$

Bellman Expectation Equation (Forma Matriciale)

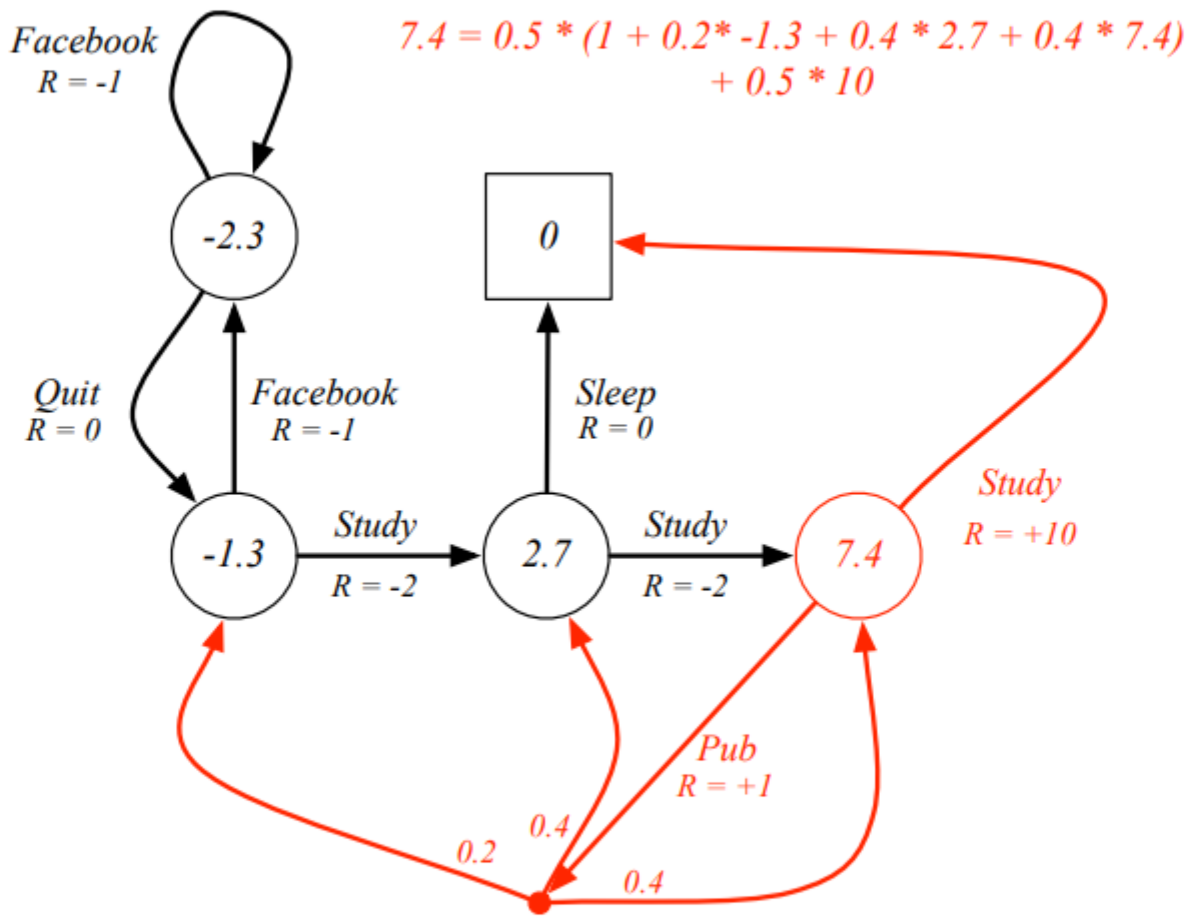
- La Bellman Expectation Equation può essere espressa in modo conciso utilizzando il MRP indotto,

$$v_\pi = \mathcal{R}^\pi + \gamma \mathcal{P}^\pi v_\pi$$

- con soluzione diretta,

$$v_\pi = (I - \gamma \mathcal{P}^\pi)^{-1} \mathcal{R}^\pi$$

Esempio: Bellman Expectation Equation per il MDP degli studenti



Optimal Value function

- La *optimal state-value function* $v_*(s)$ è la massima value function tra tutte le policy

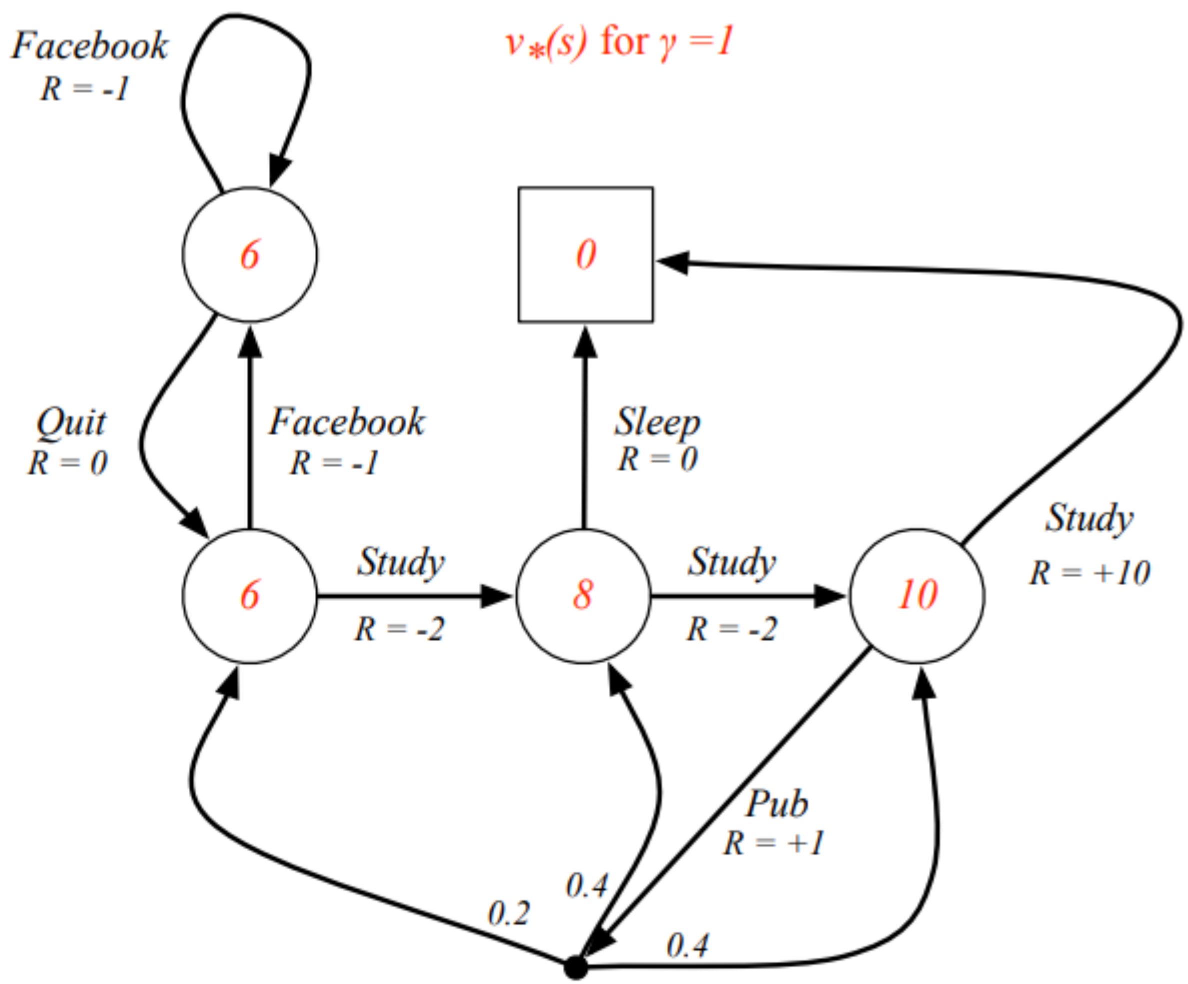
$$v_*(s) = \max_{\pi} v_\pi(s)$$

- La *optimal action-value function* $q_*(s, a)$ è la massima action-value function tra tutte le policy

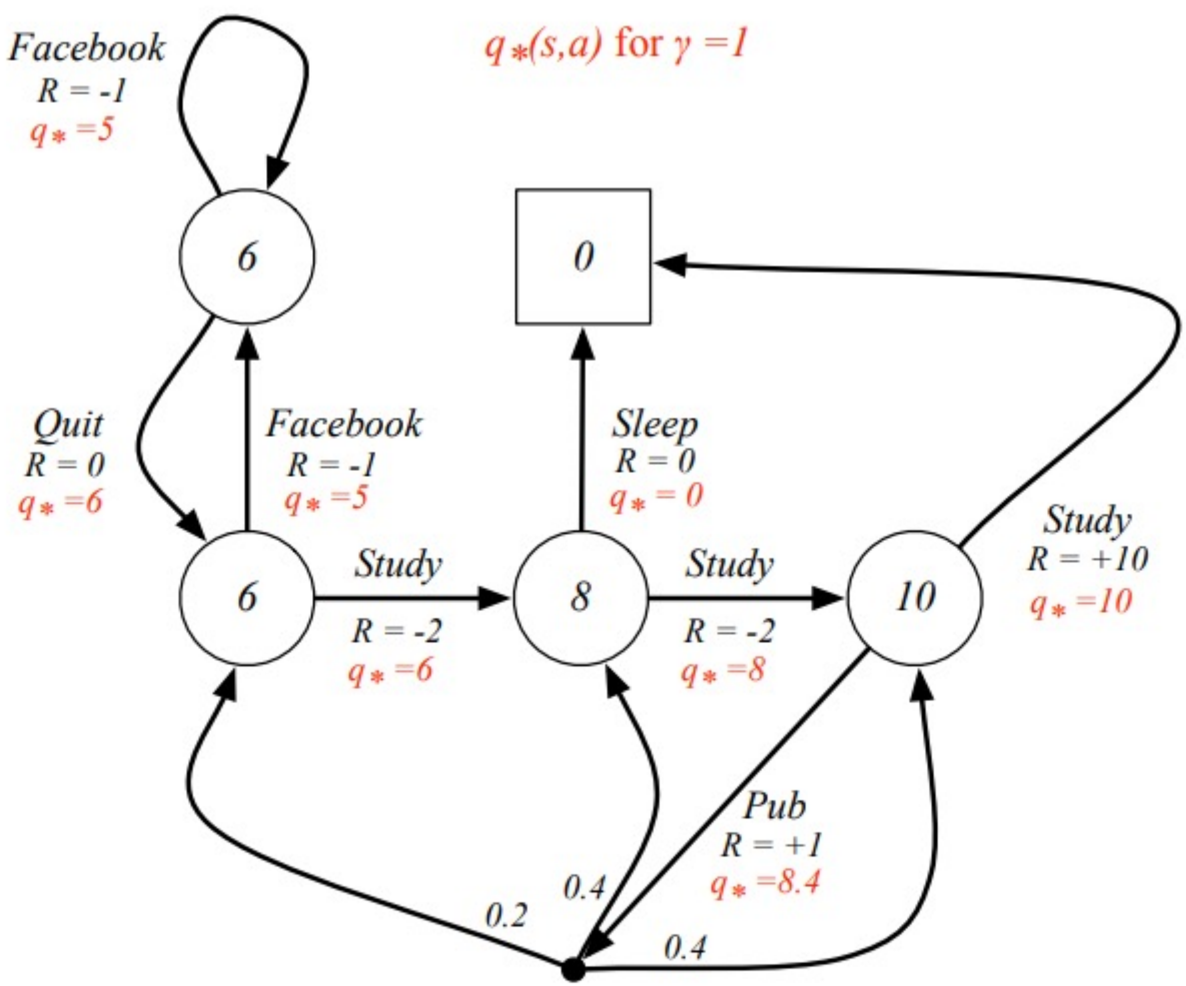
$$q_*(s, a) = \max_{\pi} q_\pi(s, a)$$

- La optimal value function specifica la migliore performance nel MDP
- Un MDP è "risolto" quando si determina l'optimal value function

Esempio: Optimal Value Function per il MDP degli studenti



Esempio: Optimal Action-Value Function per il MDP degli studenti



Policy ottimale

- Definiamo un ordinamento parziale delle policy

$$\pi \geq \pi' \text{ if } v_{\pi}(s) \geq v_{\pi'}(s), \forall s$$

- Teorema:
  - Per ogni processo decisionale di Markov
    - Esiste una policy ottimale π_* che è migliore o uguale a tutte le altre policy, $\pi_* \geq \pi, \forall \pi$
    - Tutte le policy ottimali raggiungono la optimal value function, $v_{\pi_*}(s) = v_*(s)$
    - Tutte le policy ottimali raggiungono la optimal action-value function, $q_{\pi_*}(s, a) = q_*(s, a)$

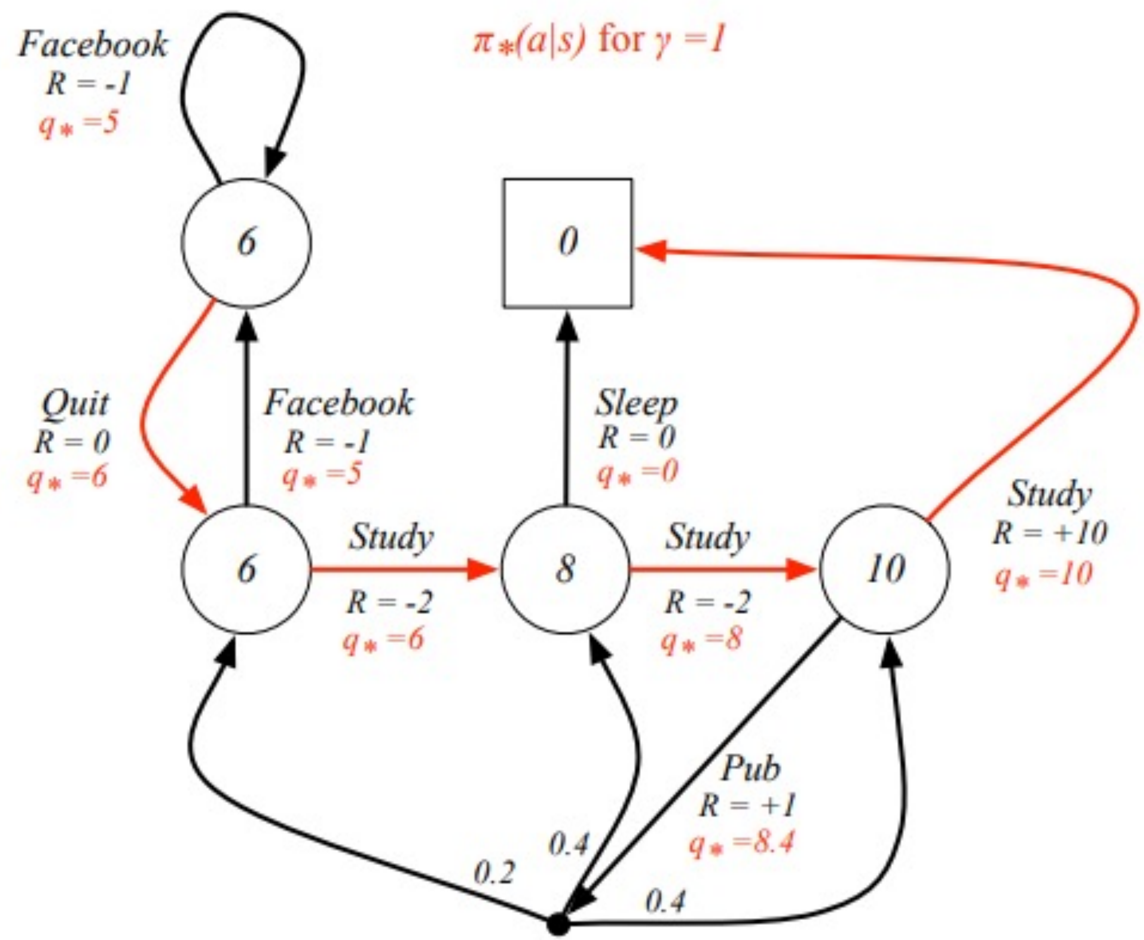
Individuare una policy ottimale

- Una policy ottimale può essere individuata massimizzando $q_*(s, a)$,

$$\pi_*(a|s) = \begin{cases} 1 & \text{if } a = \underset{a \in \mathcal{A}}{\operatorname{argmax}}\ q_*(s,a) \\ 0 & \textit{otherwise} \end{cases}$$

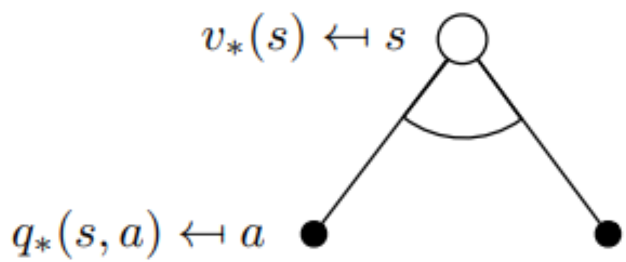
- Esiste sempre una policy ottimale deterministica per ogni MDP
- Se $q_*(s, a)$ è noto, si ottiene immediatamente la policy ottimale

Esempio: Policy ottimale per il MDP degli studenti



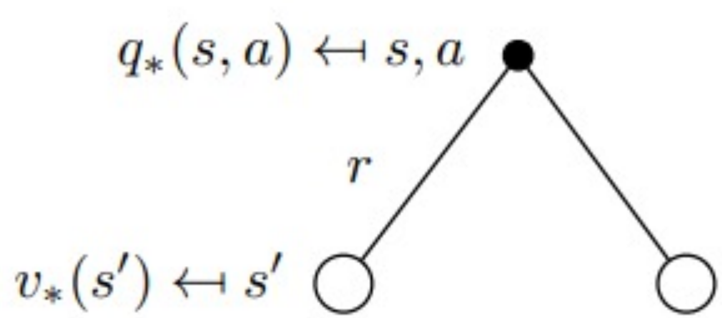
Bellman Optimality Equation per v_*

- Le optimal value functions sono ricorsivamente correlate dalle Bellman optimality equations:



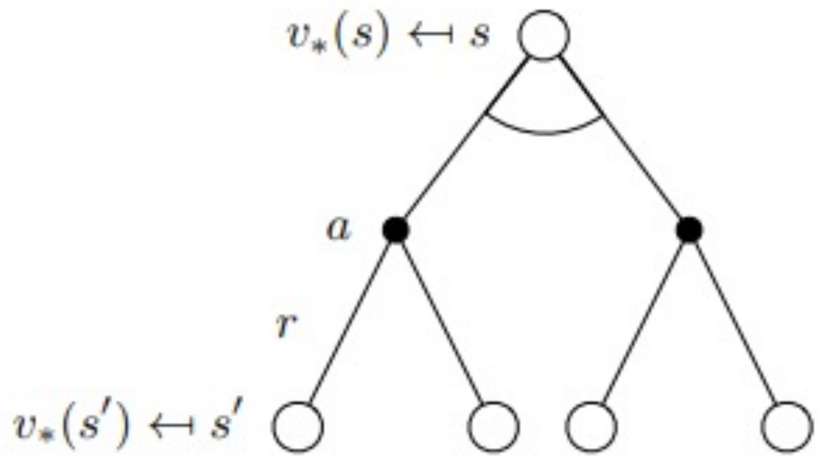
$$v_*(s) = \max_a q_*(s,a)$$

Bellman Optimality Equation per q_*



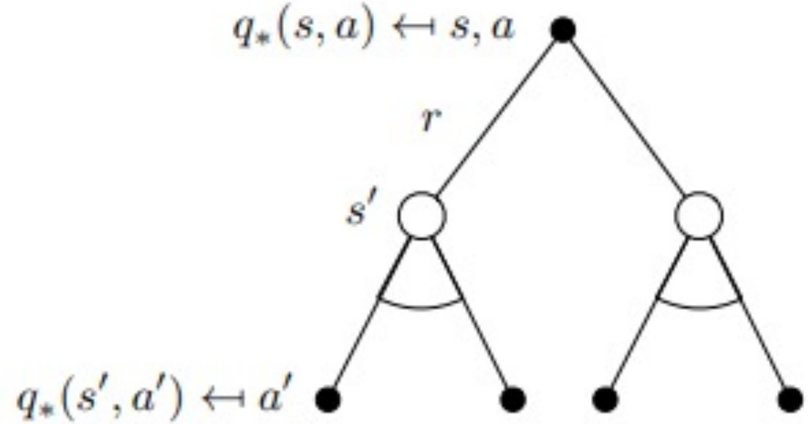
$$q_*(s,a) = \mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a v_*(s')$$

Bellman Optimality Equation per v_*(2)



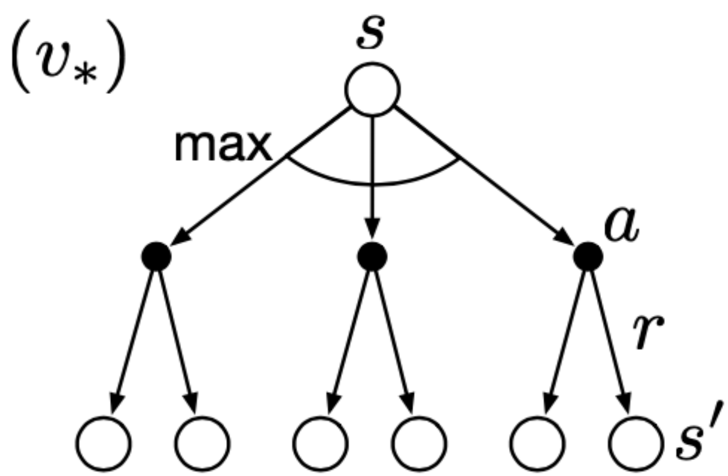
$$v_*(s) = \max_a \mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a v_*(s')$$

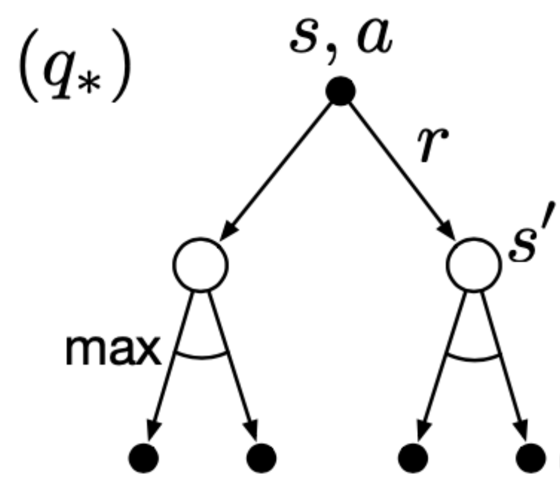
Bellman Optimality Equation per q_*(2)



$$q_*(s,a) = \mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a \max_{a'} q_*(s',a')$$

Backup diagrams for v_* and q_*

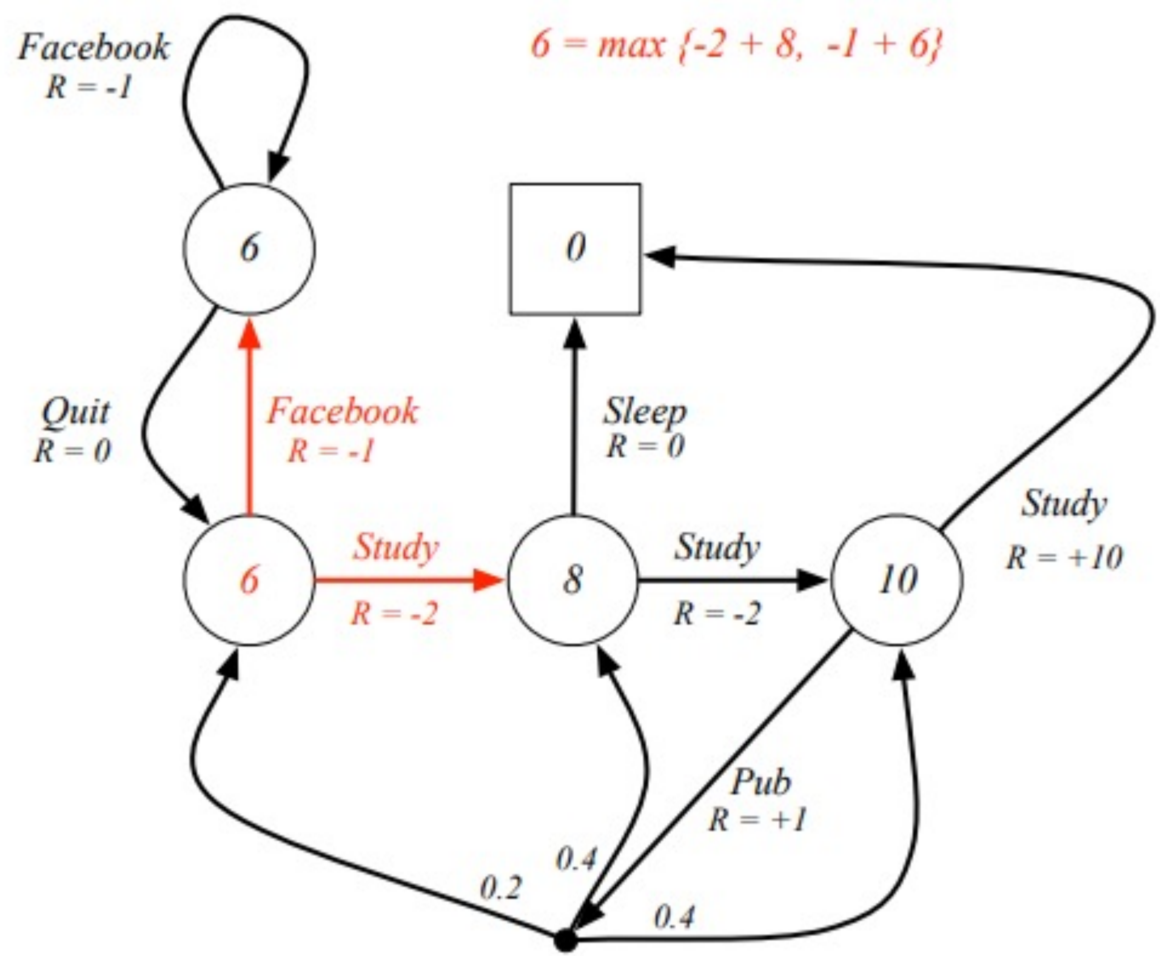




$$v_*(s) = \max_{a \in \mathcal{A}} \mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} P_{ss'}^a v_*(s')$$

$$q_*(s,a) = \mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} P_{ss'}^a \max_{a' \in \mathcal{A}} q_*(s', a')$$

Esempio: Bellman Optimality Equation per il MDP degli studenti



Risoluzione Bellman Optimality Equation

- La Bellman Optimality Equation non è lineare
- Nessuna soluzione in forma chiusa (in generale)
- La complessità computazionale è O(n^3) per *n* stati
- Esistono molti metodi iterativi per la risoluzione
  - Value Iteration
  - Policy Iteration
  - Q-learning
  - Sarsa

Estensioni del MDP

- MDP parzialmente osservabili

POMDP

- Un Processo decisionale di Markov parzialmente osservabile è un MDP composto da stati nascosti. Esso è un hidden Markov model con azioni.
- Un POMDP è una tupla *<S, A, O, P, R, Z, γ>*
  - S è un insieme finito di stati
  - A è un insieme finito di azioni
  - O è un insieme finito di osservazioni
  - P è una matrice di probabilità di transizione di stato

$$P_{ss'}^{a} = \mathrm{P}[S_{t+1} = s' \mid S_t = \mathrm{s}, A_t = a]$$

  - R è una funzione di ricompensa,

$$R_s^a = \mathrm{E}[R_{t+1} \mid S_t = \mathrm{s}, A_t = a]$$

  - Z è una funzione di osservazione,

$$Z_{s'o}^{a} = \mathrm{P}[O_{t+1} = \mathrm{o} \mid S_{t+1} = s', A_t = a]$$

  - γ è un fattore di sconto, $\gamma \in [0,1]$

Stati credenza

- Una storia H_t è una sequenza di azioni, osservazioni e ricompense,

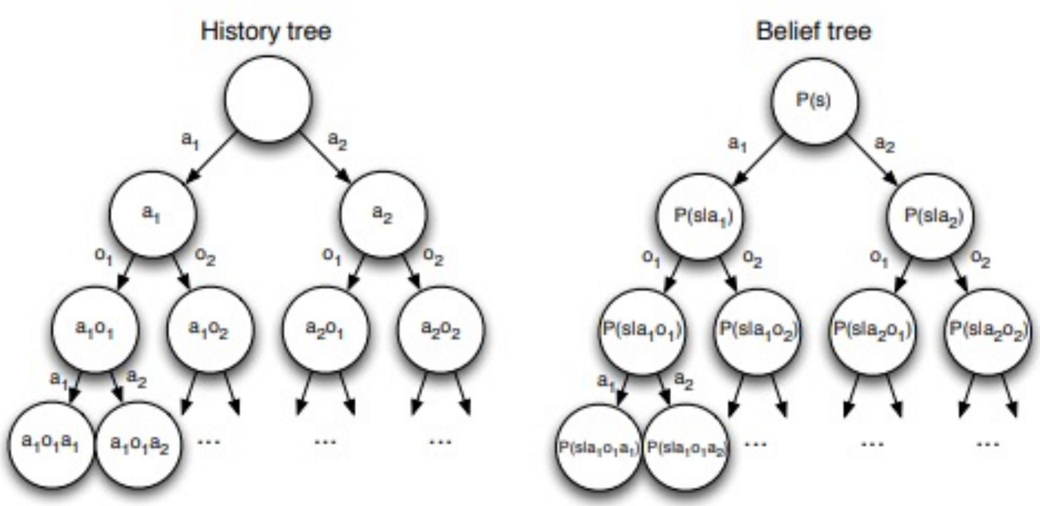
$$H_t = A_0, O_1, R_1, \ldots, A_{t-1}, O_t, R_t$$

- Uno *stato credenza b(h)* è una distribuzione di probabilità sugli stati, condizionata dalla storia *h*

$$b(h) = (\mathrm{P}[S_t = s^1 \mid H_t = h], \ldots, \mathrm{P}[S_t = s^n \mid H_t = h])$$

Riduzioni dei POMDP

- La storia H_t soddisfa la proprietà di Markov
- Lo stato credenza $b(H_t)$ soddisfa la proprietà di Markov



- Un POMDP può essere ridotto a un (infinito) albero della storia
- Un POMDP può essere ridotto a un (infinito) albero di stati credenza