



Autonomous Networking

Gaia Maselli

Dept. of Computer Science



Today's plan

Formalization of sequential decision making

- Markov Processes
- Markov Reward Processes
- Markov Decision Processes



Why MDP?



MDPs are a **classical formalization of sequential decision making**, where actions *influence* not just *immediate rewards*, but also *subsequent situations (states)* and through those future rewards



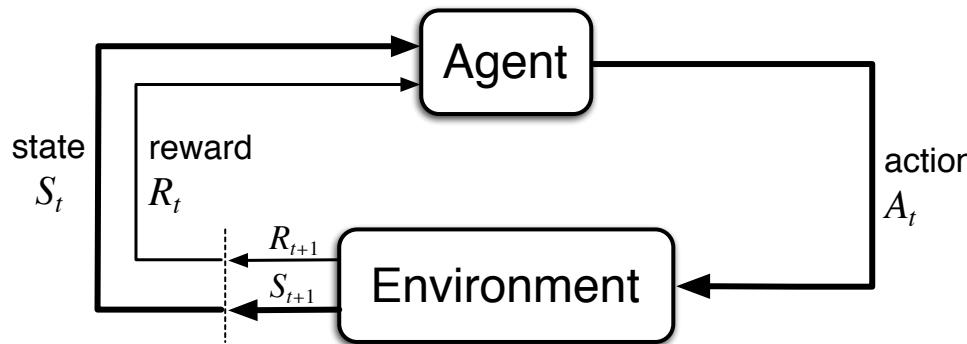
MDPs involve **delayed** rewards and the need to trade off immediate and delayed rewards



Whereas in bandit we estimated the $q^*(a)$ of each action a , in MDPs **we estimate the value $q^*(a,s)$ of each action a in each state s , or we estimate the value $v^*(s)$ of each state s given optimal action selection**

The agent-environment interface

- MDPs are meant to be a straightforward framing of the problem of learning from interaction to achieve a goal.



- The agent and environment interact at each of a sequence of discrete time steps, $t = 0, 1, 2, 3, \dots$
- At each timestep, the agent receives some representation of the environment state and on that basis selects an action
- One time step later, in part as a consequence of its action, the agent receives a numerical reward and finds itself in a new state



Introduction to MDP

- Markov decision processes formally describe an environment for reinforcement learning
- Where the environment is *fully observable*
- i.e. The current state completely characterises the process
- Almost all RL problems can be formalised as MDPs,
 - e.g. Bandits are MDPs with one state



Markov property

“The future is independent of the past given the present”

Definition

A state S_t is Markov if and only if $\mathbb{P}[S_{t+1} | S_t] = \mathbb{P}[S_{t+1} | S_1, \dots, S_t]$

- The state captures all relevant information from the history
- Once the state is known, the history may be thrown away
- i.e. The state is a sufficient statistic of the future



State Transition Matrix

- For a Markov state s and successor state s' , the *state transition probability* is defined by
- $P_{ss'} = \mathbb{P} [S_{t+1}=s' \mid S_t=s]$
- State transition matrix P defines transition probabilities from all states s to all successor states s'

$$P = \text{from} \begin{bmatrix} P_{11} & \dots & P_{1n} \\ \vdots & & \\ P_{n1} & \dots & P_{nn} \end{bmatrix} \text{to}$$

- where each row of the matrix sums to 1.



Markov process

- A Markov process is a memoryless random process, i.e. a sequence of random states S_1, S_2, \dots with the Markov property.

Definition

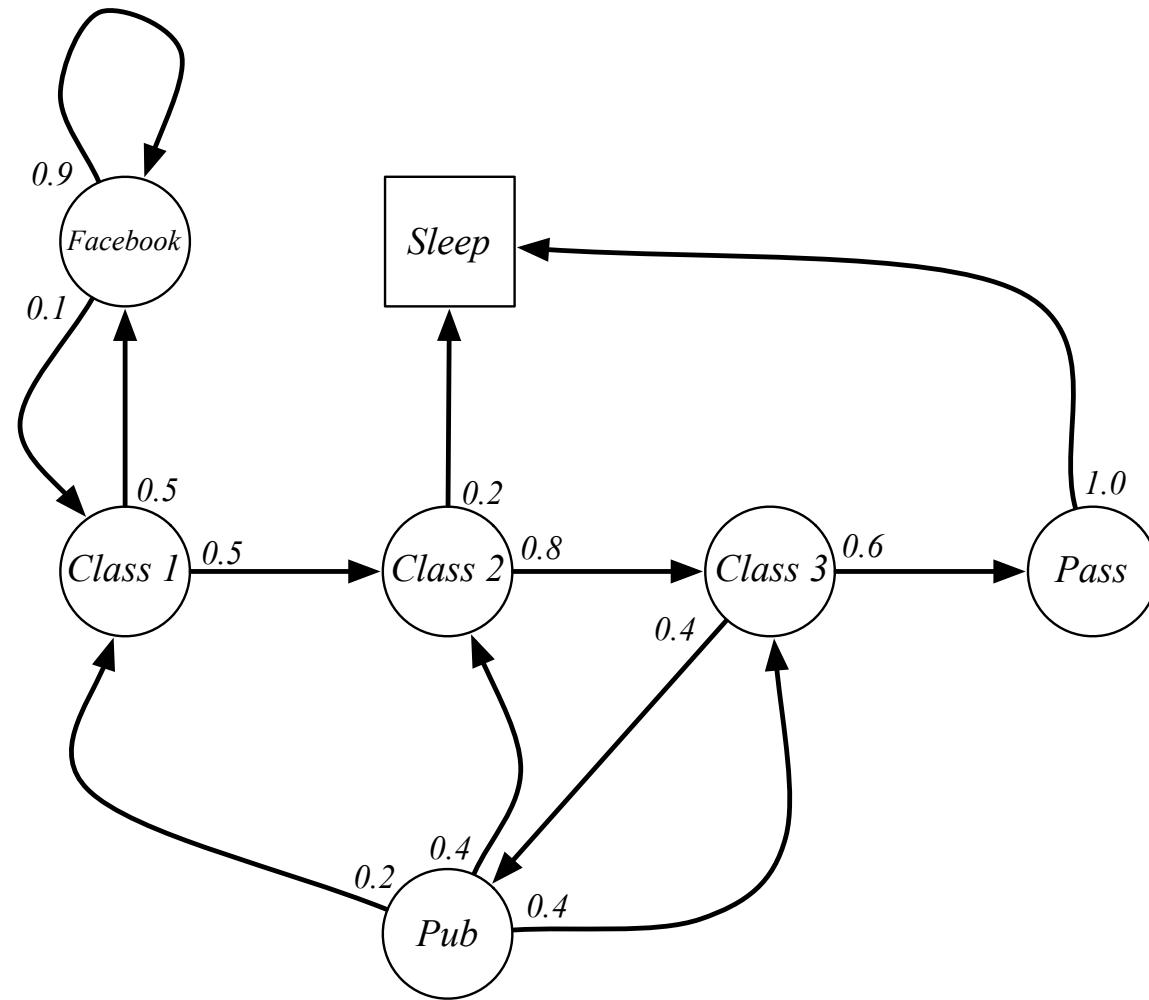
A Markov Process (or Markov Chain) is a tuple $\langle S, P \rangle$

- S is a (finite) set of states
- P is a state transition probability matrix, $P_{ss'} = \mathbb{P} [S_{t+1}=s' \mid S_t=s]$

Example: Student Markov Chain



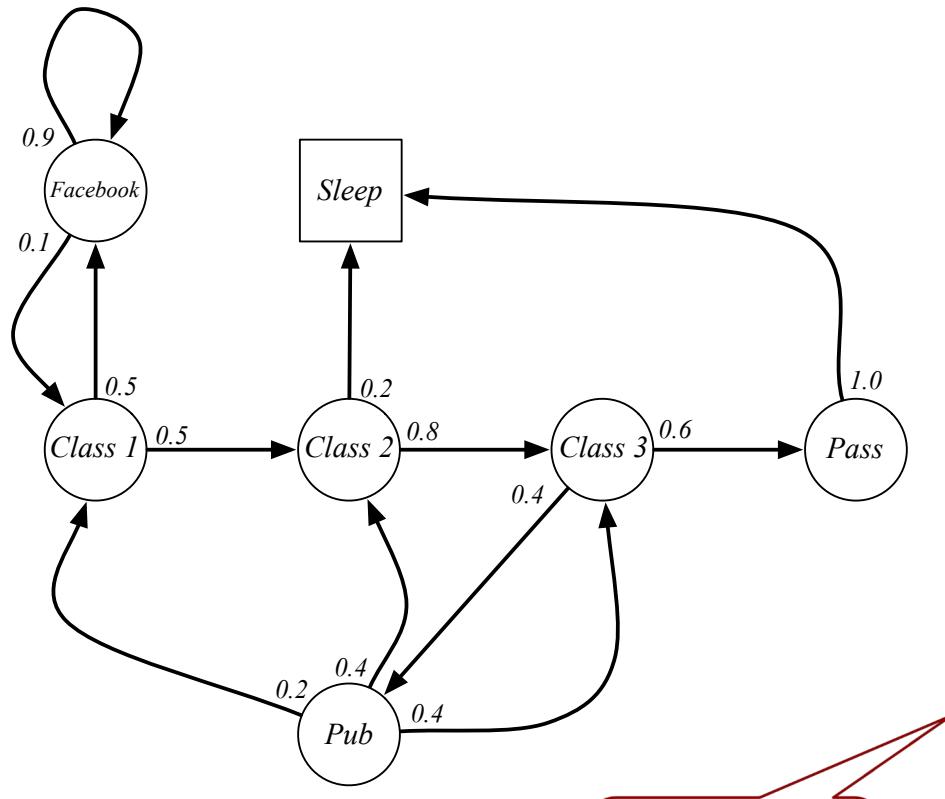
SAPIENZA
UNIVERSITÀ DI ROMA



Example: Student Markov Chain Episodes



SAPIENZA
UNIVERSITÀ DI ROMA



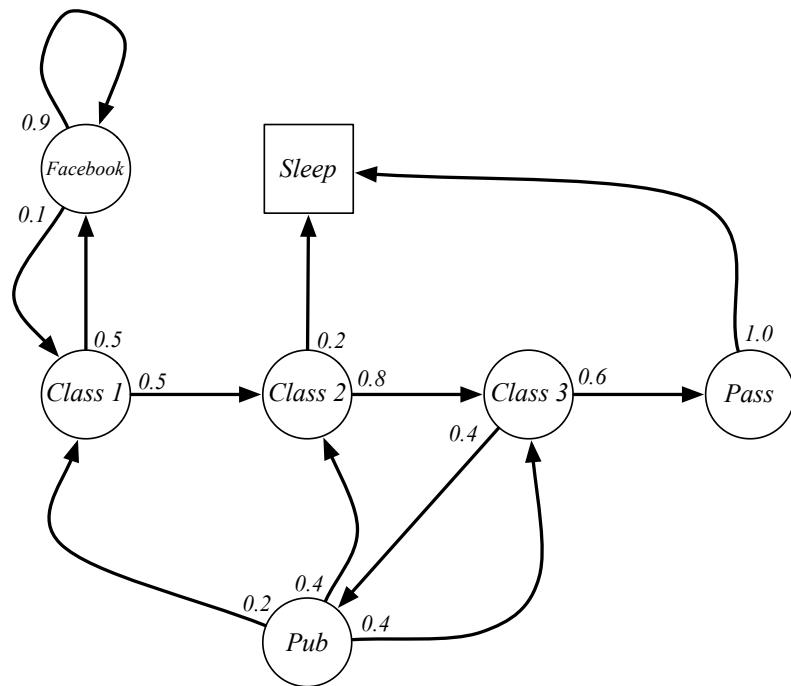
Random
sequences drawn
from probabilities

Sample **episodes** for Student Markov Chain starting from $S_1 = C_1$

S_1, S_2, \dots, S_T

- C1 C2 C3 Pass Sleep
- C1 FB FB C1 C2 Sleep
- C1 C2 C3 Pub C2 C3 Pass Sleep
- C1 FB FB C1 C2 C3 Pub C1 FB FB FB C1 C2 C3 Pub C2 Sleep

Example: Student Markov Chain Transition Matrix



$$\mathcal{P} = \begin{matrix} C1 & C2 & C3 & Pass & Pub & FB & Sleep \\ \hline C1 & 0.5 & 0.8 & 0.6 & 0.4 & 0.2 & 0.1 \\ C2 & 0.2 & 0.4 & 0.4 & 0.4 & 0.4 & 0.9 \\ C3 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 \\ Pass & 0.6 & 0.4 & 0.4 & 0.4 & 0.4 & 0.4 \\ Pub & 0.4 & 0.4 & 0.4 & 0.4 & 0.4 & 0.4 \\ FB & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 \\ Sleep & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 \end{matrix}$$

Example: recycling robot

Outline



SAPIENZA
UNIVERSITÀ DI ROMA

Formalization of sequential decision making

1. Markov Processes

- We have seen the basics on Markov processes but we have not talked about RL

2. Markov Reward Processes

- Let us add rewards to our process
- How much reward do I accumulate across a particular sequence

3. Markov Decision Processes



Markov Reward Process

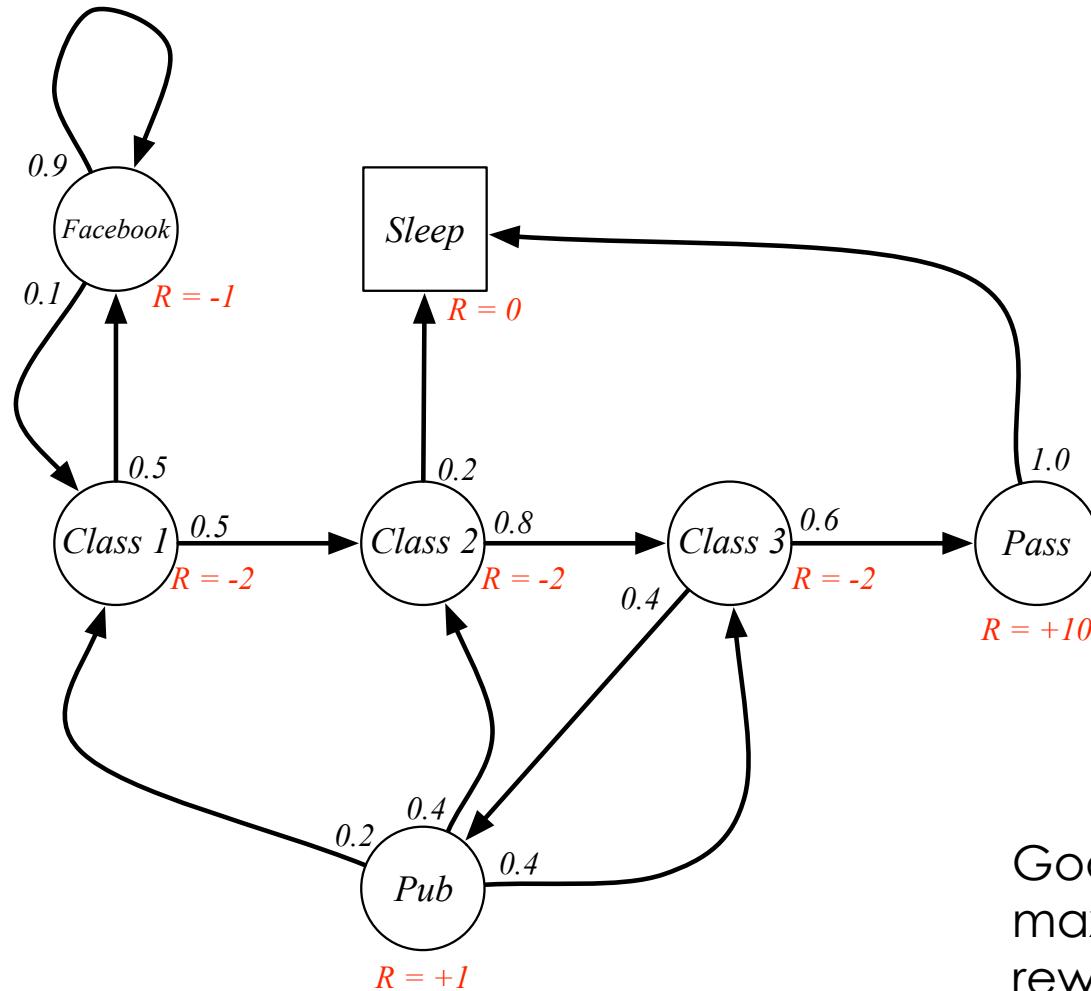
- A Markov reward process is a Markov chain with values.

Definition

A Markov Reward Process is a tuple $\langle S, P, R, \gamma \rangle$

- S is a (finite) set of states
- P is a state transition probability matrix, $P_{ss'} = \mathbb{P} [S_{t+1}=s' \mid S_t=s]$
- R is a reward function, $R_s = \mathbb{E} [R_{t+1} \mid S_t = s]$
- γ is a discount factor, $\gamma \in [0, 1]$

Example: Student MRP



Goal: we want to maximize the rewards we obtain

Return



Definition

The return G_t is the total discounted reward from time-step t .

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

- The discount $\gamma \in [0, 1]$ is the present value of future rewards
- The value of receiving reward R after $k + 1$ time-steps is $\gamma^k R$
- This values immediate reward above delayed reward
 - γ close to 0 leads to "short-sighted" evaluation
 - γ close to 1 leads to "far-sighted" evaluation



Why discount?

Most Markov reward and decision processes are discounted. Why?

- Mathematically convenient to discount rewards
- Avoids infinite returns in cyclic Markov processes
- Uncertainty about the future may not be fully represented
- If the reward is financial, immediate rewards may earn more interest than delayed rewards
- Animal/human behaviour shows preference for immediate reward
- It is sometimes possible to use undiscounted Markov reward processes (i.e. $\gamma = 1$), e.g. if all sequences terminate.



Value function

- The value function $v(s)$ gives the long-term value of state s

Definition

The state value function $v(s)$ of an MRP is the **expected return starting from state s**

$$V_s = \mathbb{E} [G_t | St = s]$$

Example: Student MRP Returns



- Sample returns for Student MRP:

Starting from $S_1 = C1$ with $\gamma = \frac{1}{2}$

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

C1 C2 C3 Pass Sleep

C1 FB FB C1 C2 Sleep

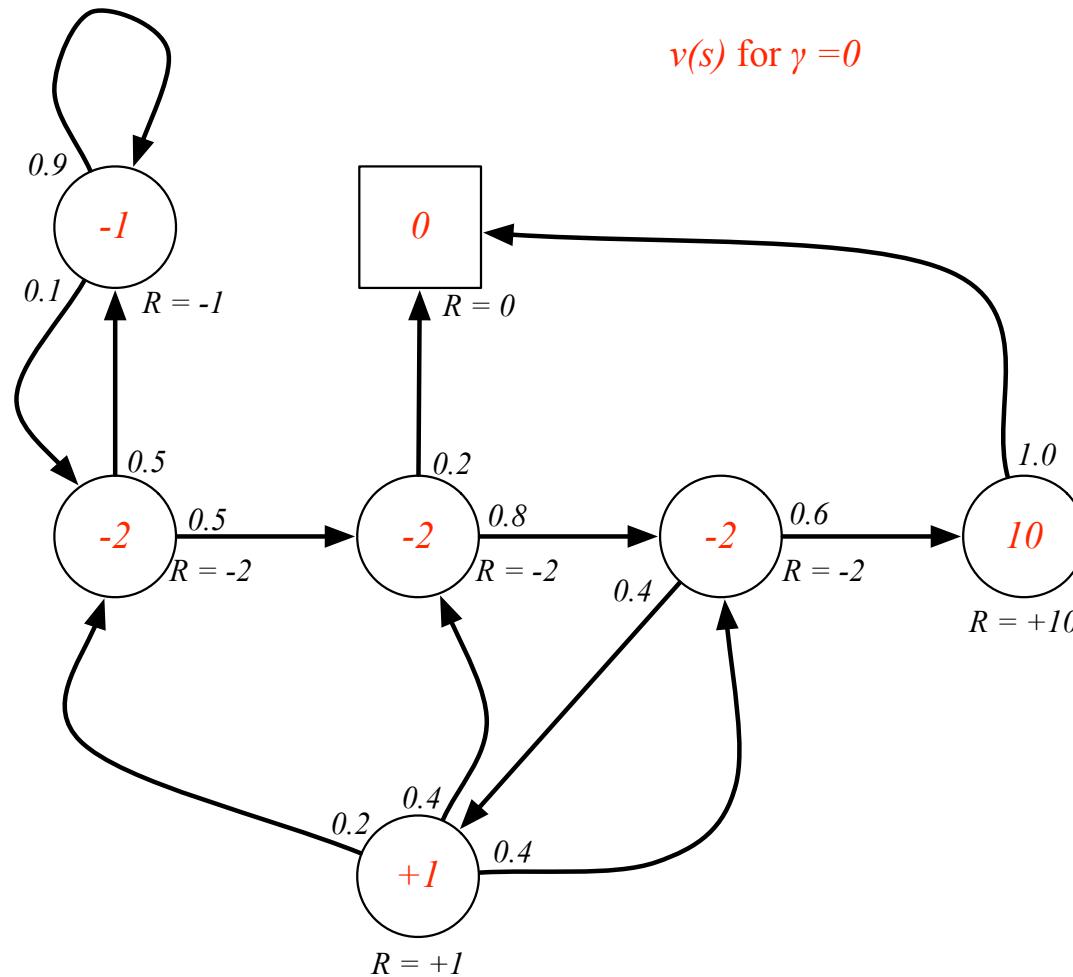
C1 C2 C3 Pub C2 C3 Pass Sleep

C1 FB FB C1 C2 C3 Pub C1 ...

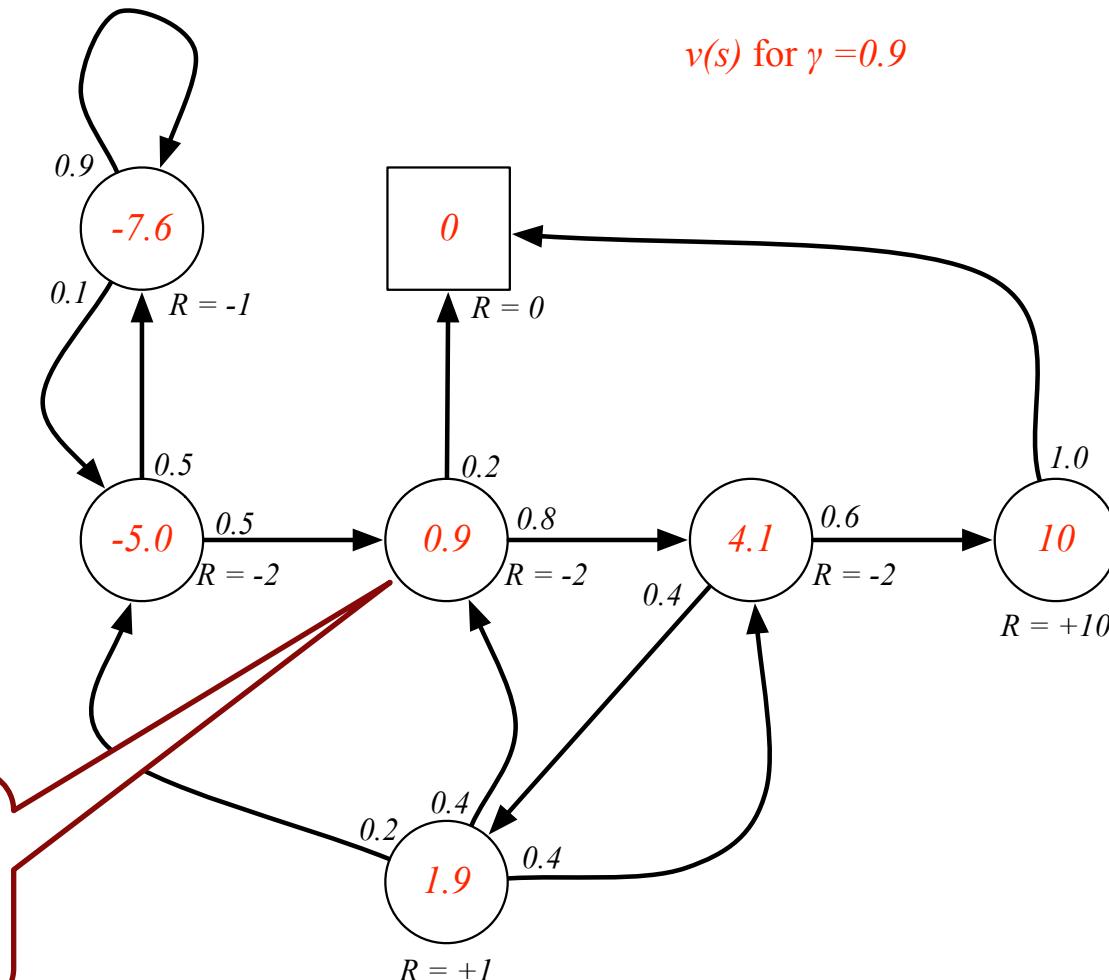
FB FB FB C1 C2 C3 Pub C2 Sleep

$$\begin{aligned} v_1 &= -2 - 2 * \cancel{\frac{1}{2}} - 2 * \cancel{\frac{1}{4}} + 10 * \cancel{\frac{1}{8}} &= -2.25 \\ v_1 &= -2 - 1 * \frac{1}{2} - 1 * \frac{1}{4} - 2 * \frac{1}{8} - 2 * \frac{1}{16} &= -3.125 \\ v_1 &= -2 - 2 * \frac{1}{2} - 2 * \frac{1}{4} + 1 * \frac{1}{8} - 2 * \frac{1}{16} \dots &= -3.41 \\ v_1 &= -2 - 1 * \frac{1}{2} - 1 * \frac{1}{4} - 2 * \frac{1}{8} - 2 * \frac{1}{16} \dots &= -3.20 \end{aligned}$$

Example: State-Value Function for Student MRP (1)



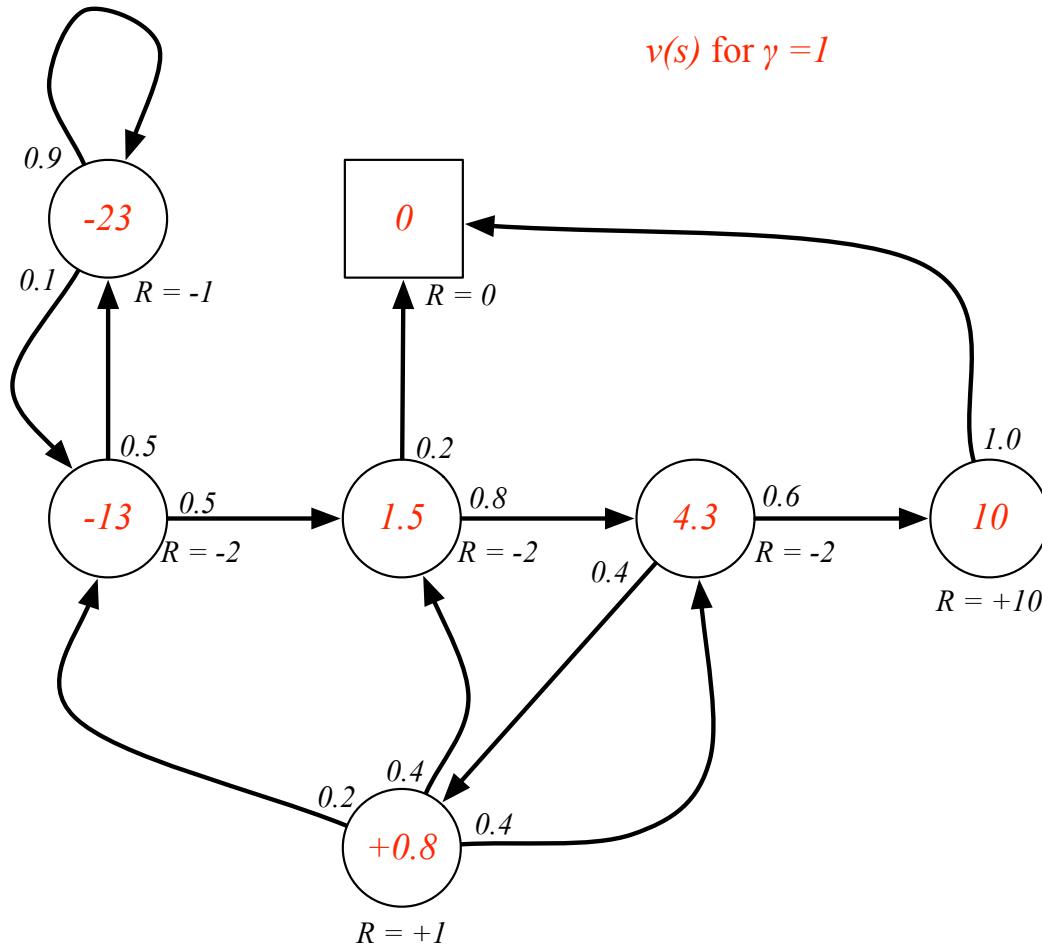
Example: State-Value Function for Student MRP (2)



Example: State-Value Function for Student MRP (3)



SAPIENZA
UNIVERSITÀ DI ROMA



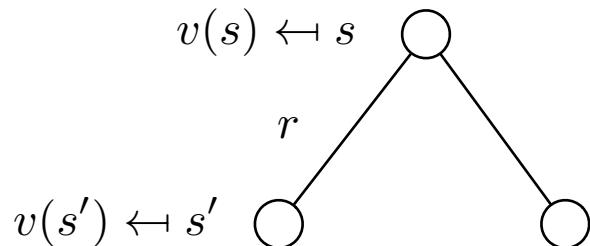
Bellman Equation for MRPs

- The value function can be decomposed into two parts:
 - immediate reward R_{t+1}
 - discounted value of successor state $\gamma v(S_{t+1})$

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

Bellman Equation for MRPs (2)

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

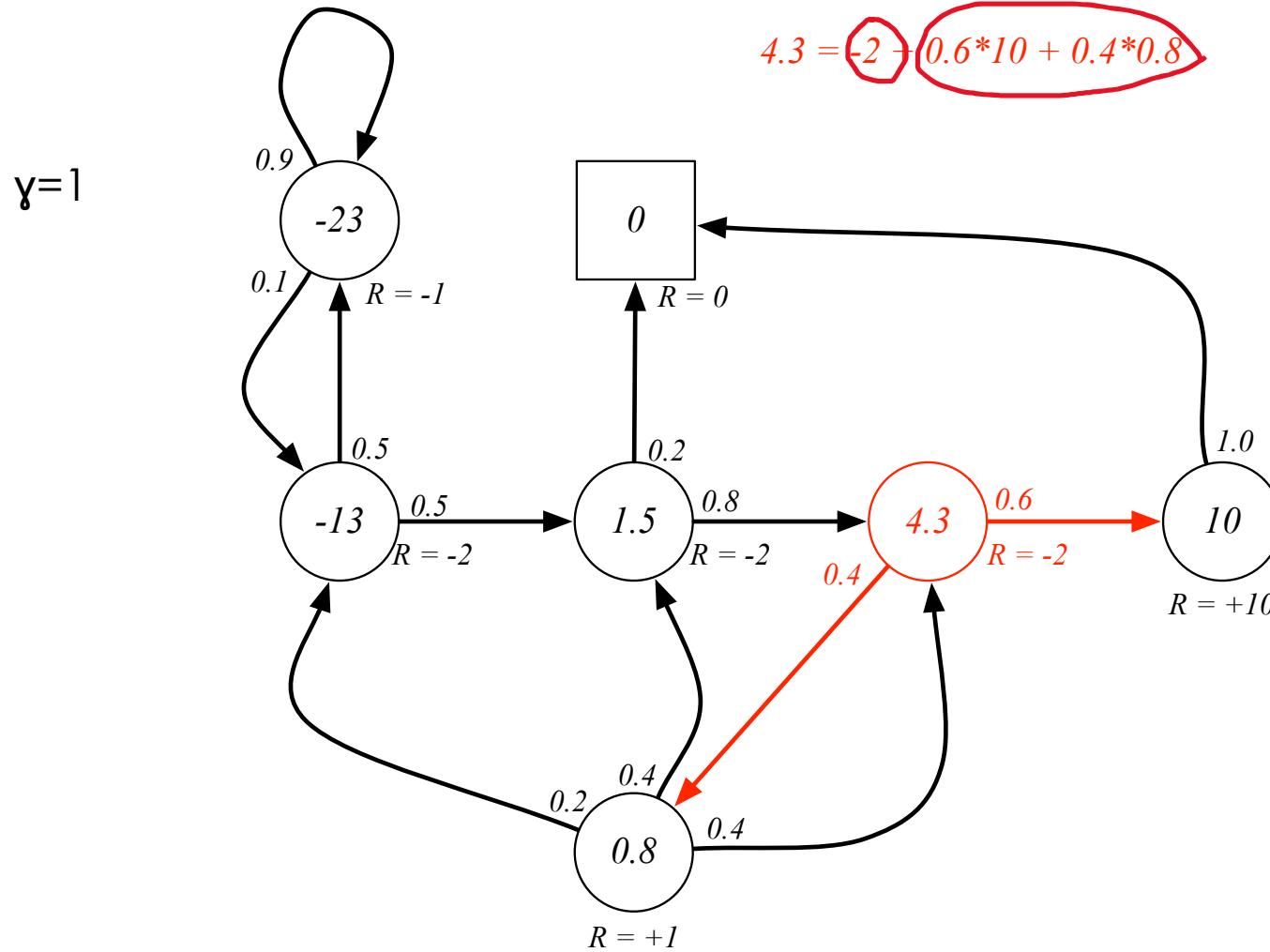


- Bellman equations expresses a relationship between the value of a state and the values of its successor states

$$v(s) = \mathcal{R}_s + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'} v(s')$$

- Bellman equation **averages over all the possibilities, weighting each by its probability of occurring**
- The value of the start state must be equal the (discounted) value of the expected next state, plus the reward expected along the way

Example: Bellman Equation for Student MRP



Bellman Equation in Matrix Form



- The Bellman equation can be expressed concisely using matrices,

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

- where v is a column vector with one entry per state

$$\begin{bmatrix} v(1) \\ \vdots \\ v(n) \end{bmatrix} = \begin{bmatrix} \mathcal{R}_1 \\ \vdots \\ \mathcal{R}_n \end{bmatrix} + \gamma \begin{bmatrix} \mathcal{P}_{11} & \dots & \mathcal{P}_{1n} \\ \vdots & & \\ \mathcal{P}_{n1} & \dots & \mathcal{P}_{nn} \end{bmatrix} \begin{bmatrix} v(1) \\ \vdots \\ v(n) \end{bmatrix}$$



Solving the Bellman Equation

- The Bellman equation is a linear equation
- It can be solved directly:

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

- Computational complexity is $O(n^3)$ for n states
- Direct solution only possible for small MRPs
- There are many iterative methods for large MRPs, e.g.
 - Dynamic programming
 - Monte-Carlo evaluation
 - Temporal-Difference learning

Outline



SAPIENZA
UNIVERSITÀ DI ROMA

Formalization of sequential decision making

1. Markov Processes
2. Markov Reward Processes
3. Markov Decision Processes



Markov Decision Process

- A Markov decision process (MDP) is a Markov reward process with decisions. It is an *environment* in which all states are Markov

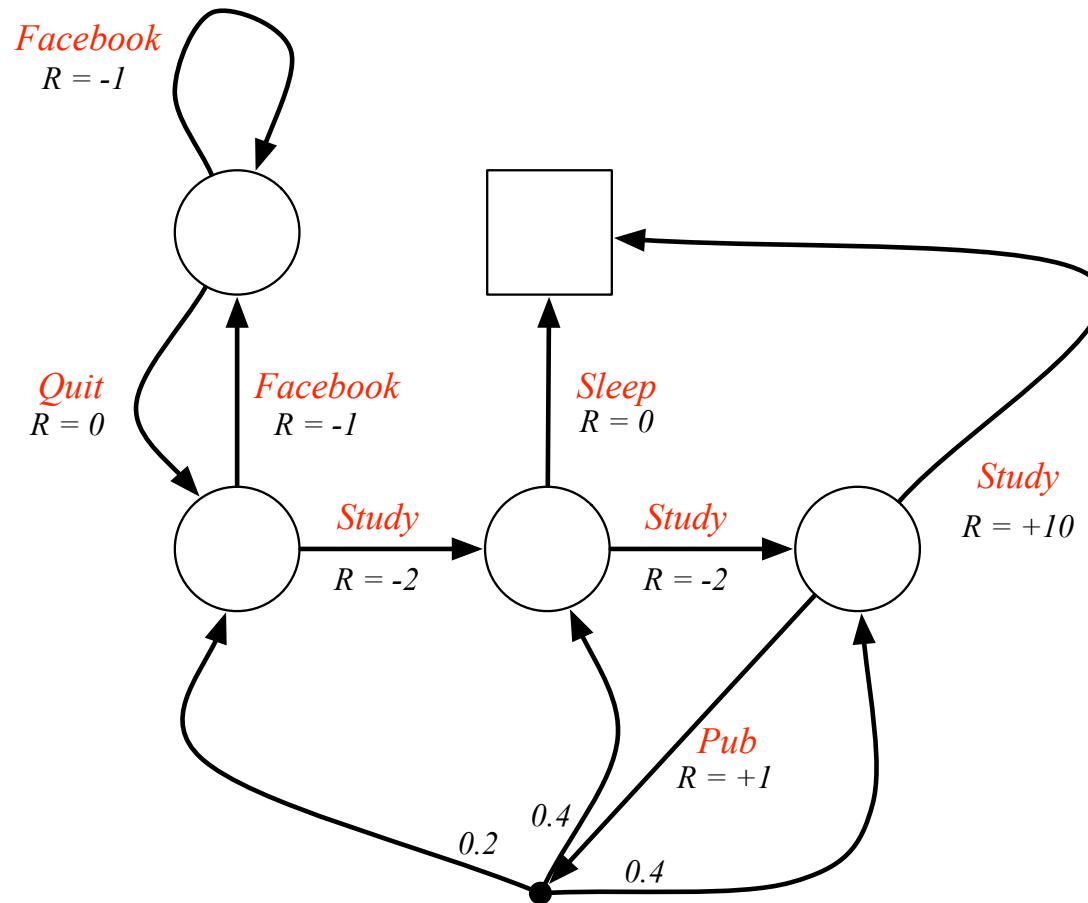
Definition

A Markov Reward Process is a tuple $\langle S, A, P, R, \gamma \rangle$

- S is a (finite) set of states
- A is a finite set of actions
- P is a state transition probability matrix,
$$P_{ss'}^{\textcolor{red}{a}} = \mathbb{P} [S_{t+1} = s' \mid S_t = s, A_t = \textcolor{red}{a}]$$
- R is a reward function,
$$R_s^{\textcolor{red}{a}} = \mathbb{E} [R_{t+1} \mid S_t = s, A_t = \textcolor{red}{a}]$$
- γ is a discount factor, $\gamma \in [0, 1]$

One matrix
for each
action

Example: Student MDP





Policies (1)

Definition

A policy is a distribution over actions given states

$$\pi(a | s) = \mathbb{P} [A_t=a \mid S_t=s]$$

- A policy fully defines the behaviour of an agent
- MDP policies depend on the current state (not the history)
- i.e. Policies are stationary (time-independent)



Value Function

Definition

The *state-value function* $v_\pi(s)$ of an MDP is the expected return starting from state s , and then **following policy π**

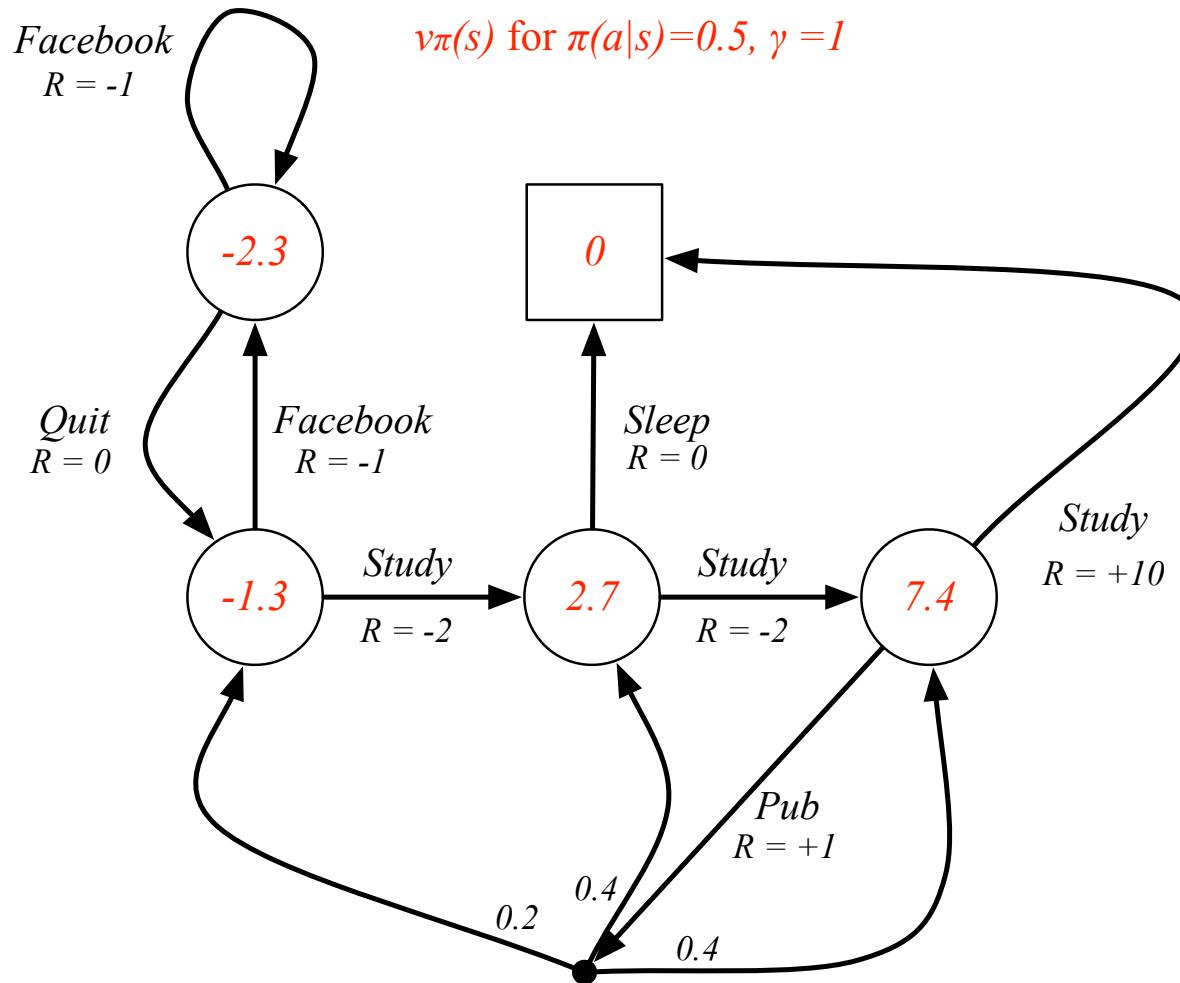
$$v_\pi(s) = \mathbb{E}_\pi [G_t \mid S_t=s]$$

Definition

The *action-value function* $q_\pi(s,a)$ is the expected return starting from state s , **taking action a , and then following policy π**

$$q_\pi(a \mid s) = \mathbb{E}_\pi [G_t \mid S_t=s, A_t=a]$$

Example: State-Value Function for Student MDP



Bellman Expectation Equation



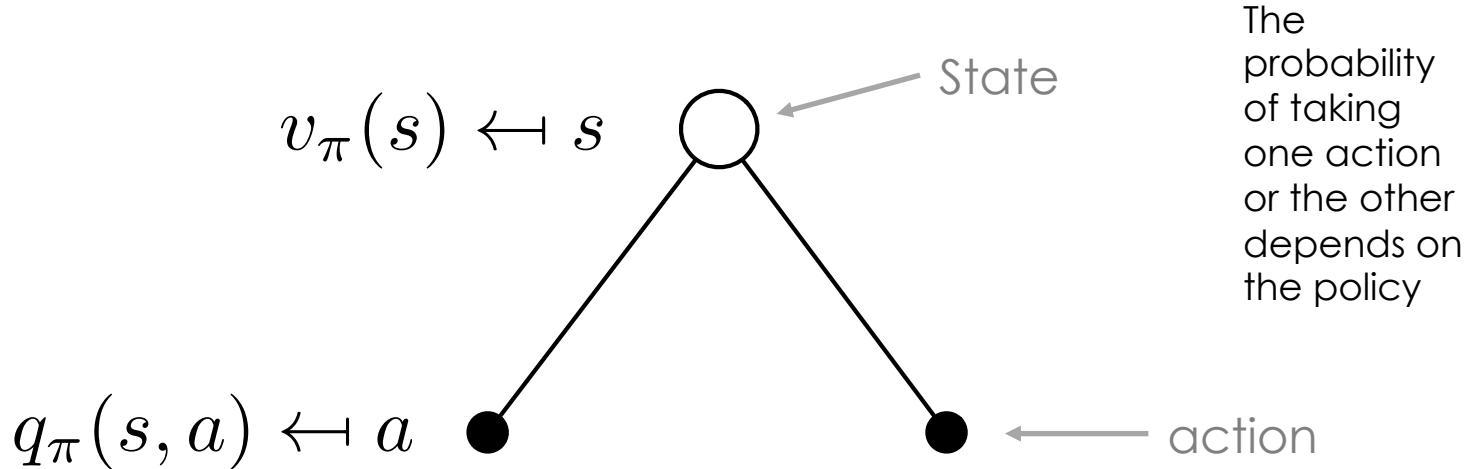
- The state-value function can again be decomposed into immediate reward plus discounted value of successor state,

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

- The action-value function can similarly be decomposed,

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

Bellman Expectation Equation for V^π

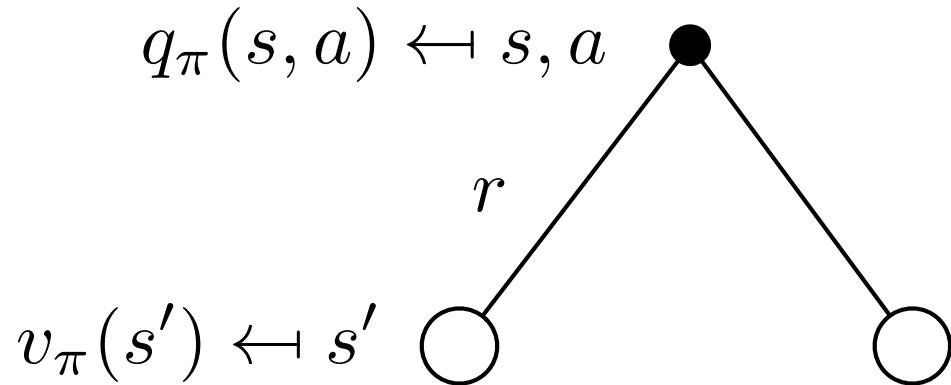


$$v_\pi(s) = \sum_{a \in \mathcal{A}} \pi(a|s) q_\pi(s, a)$$

Bellman Expectation Equation for Q^π

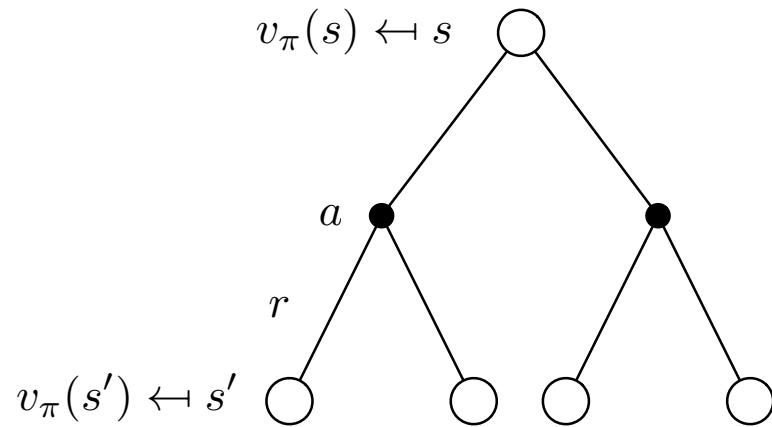


SAPIENZA
UNIVERSITÀ DI ROMA



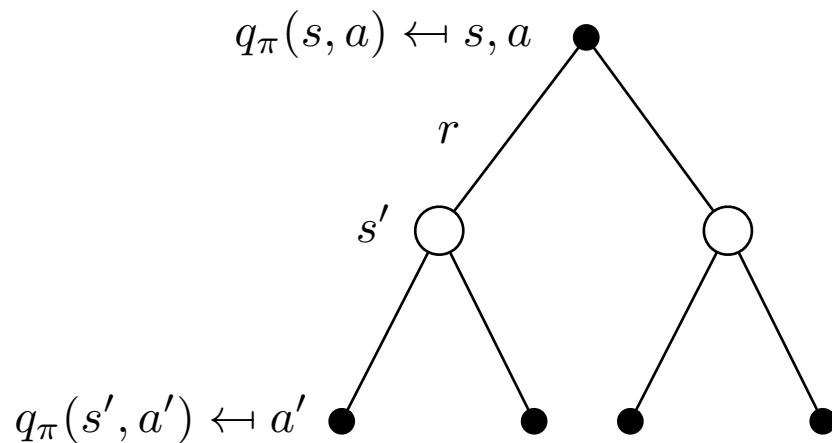
$$q_\pi(s, a) = \mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a v_\pi(s')$$

Bellman Expectation Equation for v_π (2)



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

Bellman Expectation Equation for q_{π} (2)



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

Example: Bellman Expectation Equation in Student MDP

