UPPSALA UNIVERSITET

Reinforcement Learning

Lecture 2 - Markov Decision Processes

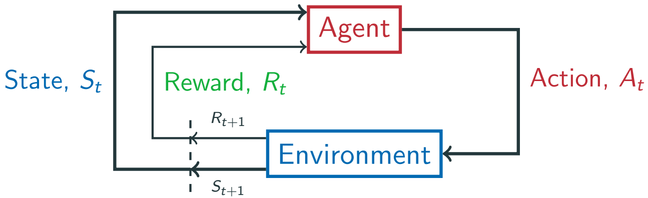
Per Mattsson

2022

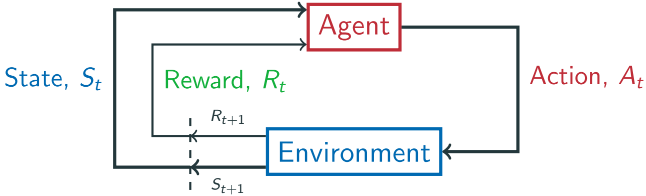
Department of Information Technology

Repetition

Repetition

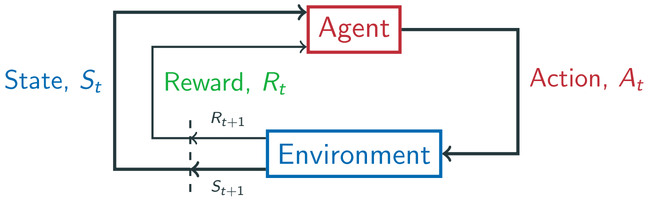


Repetition



- **Markov property:** State contains all information that is useful to predict future:

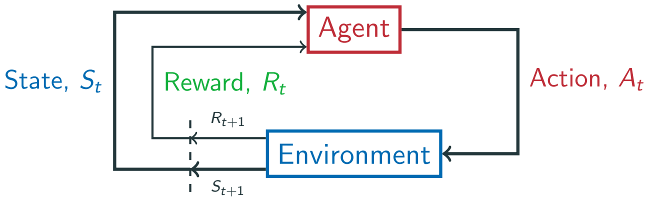
$$p(S_{t+1}|S_t, A_t, S_{t-1}, A_{t-1}, \dots, S_0, A_0) = p(S_{t+1}|S_t, A_t).$$



- **Markov property:** State contains all information that is useful to predict future:

$$p(S_{t+1}|S_t, A_t, S_{t-1}, A_{t-1}, \ldots, S_0, A_0) = p(S_{t+1}|S_t, A_t).$$

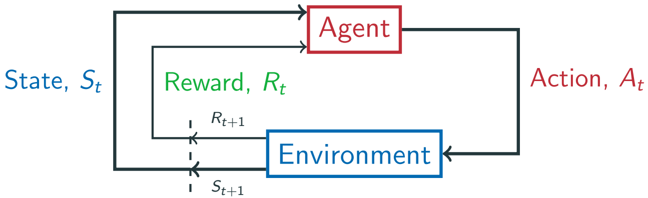
- **Policy:** $\pi(a|s)$ (probability of choosing a when we are in state s).



- **Markov property:** State contains all information that is useful to predict future:

$$p(S_{t+1}|S_t, A_t, S_{t-1}, A_{t-1}, \ldots, S_0, A_0) = p(S_{t+1}|S_t, A_t).$$

- **Policy:** $\pi(a|s)$ (probability of choosing a when we are in state s).
- **Prediction:** Following a policy, what will the future cumulative reward be?



- **Markov property:** State contains all information that is useful to predict future:

$$p(S_{t+1}|S_t, A_t, S_{t-1}, A_{t-1}, \ldots, S_0, A_0) = p(S_{t+1}|S_t, A_t).$$

- **Policy:** $\pi(a|s)$ (probability of choosing a when we are in state s).
- **Prediction:** Following a policy, what will the future cumulative reward be?
- **Control:** Find the policy that maximize the cumulative future reward.

Markov Decision Process (MDP)

(Finite) Markov Decision Processes (MDP)

UPPSALA UNIVERSITET

- Assume that $\mathcal{S}$, $\mathcal{A}$ and $\mathcal{R}$ all have a finite number of elements.

(Finite) Markov Decision Processes (MDP)

- Assume that $\mathcal{S}$, $\mathcal{A}$ and $\mathcal{R}$ all have a finite number of elements.
- **Transition probabilities:**

$$p(s', r|s, a) = \Pr\{S_{t+1} = s', R_{t+1} = r|S_t = s, A_t = a\}$$

(Finite) Markov Decision Processes (MDP)

- Assume that $\mathcal{S}$, $\mathcal{A}$ and $\mathcal{R}$ all have a finite number of elements.
- **Transition probabilities:**

$$p(s', r|s, a) = \Pr\{S_{t+1} = s', R_{t+1} = r|S_t = s, A_t = a\}$$

- **Markov property:** Completely determines the *dynamics* of the environment.

(Finite) Markov Decision Processes (MDP)

- Assume that $\mathcal{S}$, $\mathcal{A}$ and $\mathcal{R}$ all have a finite number of elements.
- **Transition probabilities:**

$$p(s', r|s, a) = \Pr\{S_{t+1} = s', R_{t+1} = r|S_t = s, A_t = a\}$$

- **Markov property:** Completely determines the *dynamics* of the environment.
- **The expected reward:**

$$r(s, a) = \mathbb{E}[R_{t+1}|S_t = s, A_t = a]$$

(Finite) Markov Decision Processes (MDP)

- Assume that $\mathcal{S}$, $\mathcal{A}$ and $\mathcal{R}$ all have a finite number of elements.
- **Transition probabilities:**

$$p(s', r|s, a) = \Pr\{S_{t+1} = s', R_{t+1} = r|S_t = s, A_t = a\}$$

- **Markov property:** Completely determines the *dynamics* of the environment.
- **The expected reward:**

$$r(s, a) = \mathbb{E}[R_{t+1}|S_t = s, A_t = a] = \sum_{r \in \mathcal{R}} \sum_{s' \in \mathcal{S}} r p(s', r|s, a)$$

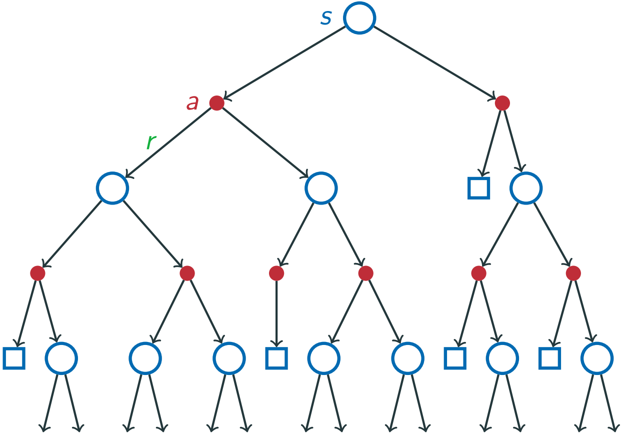
- Assume that $\mathcal{S}$, $\mathcal{A}$ and $\mathcal{R}$ all have a finite number of elements.
- **Transition probabilities:**

$$p(s', r|s, a) = \Pr\{S_{t+1} = s', R_{t+1} = r|S_t = s, A_t = a\}$$

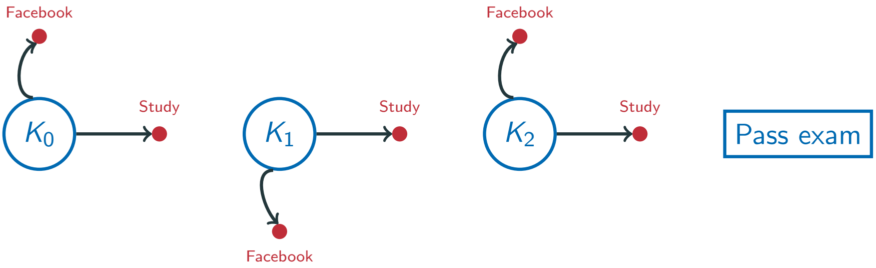
- **Markov property:** Completely determines the *dynamics* of the environment.
- **The expected reward:**

$$r(s, a) = \mathbb{E}[R_{t+1}|S_t = s, A_t = a] = \sum_{r \in \mathcal{R}} \sum_{s' \in \mathcal{S}} r p(s', r|s, a) = \sum_{r, s'} r p(s', r|s, a).$$

Markov Decision Process Tree

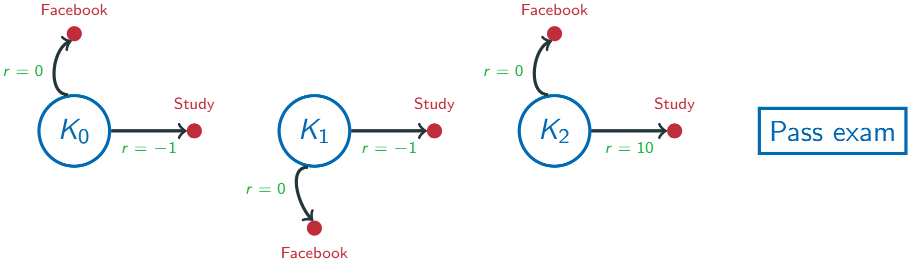


Example: Study or Facebook?

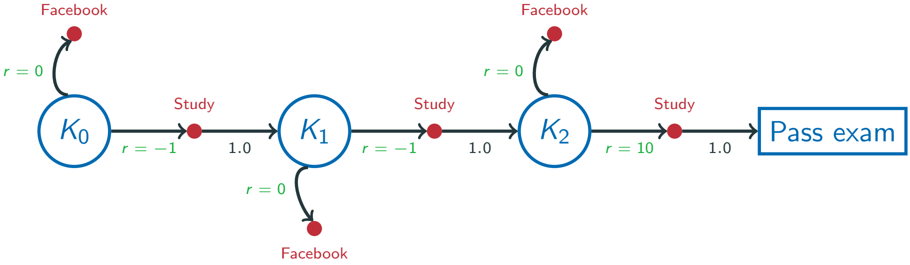


- **States:** Knowledge 0, Knowledge 1, Knowledge 2, Pass exam (terminating).
- **Actions:** Study or Facebook.

Example: Study or Facebook?

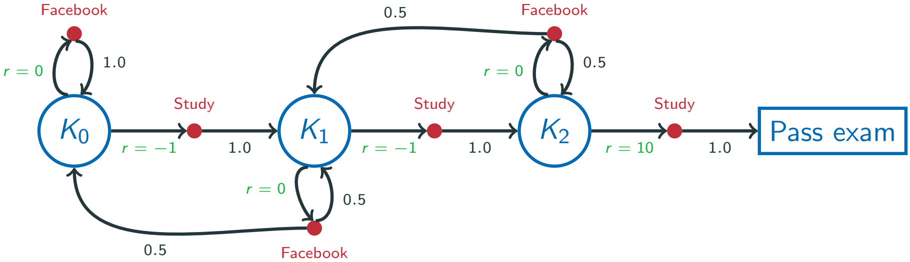


- **States:** Knowledge 0, Knowledge 1, Knowledge 2, Pass exam (terminating).
- **Actions:** Study or Facebook.



- **States:** Knowledge 0, Knowledge 1, Knowledge 2, Pass exam (terminating).
- **Actions:** Study or Facebook.

Example: Study or Facebook?



- **States:** Knowledge 0, Knowledge 1, Knowledge 2, Pass exam (terminating).
- **Actions:** Study or Facebook.

Episodic vs continuing tasks

Episodic tasks:

- Has terminating states, and the task ends in finite time.

Episodic vs continuing tasks

UPPSALA UNIVERSITET

Episodic tasks:

- Has terminating states, and the task ends in finite time.
- If you reach a terminating state the episode stops.

Episodic vs continuing tasks

UPPSALA UNIVERSITET

Episodic tasks:

- Has terminating states, and the task ends in finite time.
- If you reach a terminating state the episode stops.
- **Example:** Taxi environment or "Study or Facebook"-MDP.

Episodic vs continuing tasks

Episodic tasks:

- Has terminating states, and the task ends in finite time.
- If you reach a terminating state the episode stops.
- **Example:** Taxi environment or "Study or Facebook"-MDP.
- "If you reach a terminating state, you will stay there forever and receive no future reward."

Episodic vs continuing tasks

Episodic tasks:

- Has terminating states, and the task ends in finite time.
- If you reach a terminating state the episode stops.
- **Example:** Taxi environment or "Study or Facebook"-MDP.
- "If you reach a terminating state, you will stay there forever and receive no future reward."

Continuing tasks:

- Often not a clear way to divide up the task into independent episodes.

Episodic vs continuing tasks

Episodic tasks:

- Has terminating states, and the task ends in finite time.
- If you reach a terminating state the episode stops.
- **Example:** Taxi environment or "Study or Facebook"-MDP.
- "If you reach a terminating state, you will stay there forever and receive no future reward."

Continuing tasks:

- Often not a clear way to divide up the task into independent episodes.
- **Example:** Keep balancing the pendulum. No state where the task is done.

Episodic vs continuing tasks

Episodic tasks:

- Has terminating states, and the task ends in finite time.
- If you reach a terminating state the episode stops.
- **Example:** Taxi environment or “Study or Facebook”-MDP.
- “If you reach a terminating state, you will stay there forever and receive no future reward.”

Continuing tasks:

- Often not a clear way to divide up the task into independent episodes.
- **Example:** Keep balancing the pendulum. No state where the task is done.
- We have to take into account infinitely many future rewards.

The return

- In a given state we want to maximize future rewards $R_{t+1}, R_{t+2}, \ldots$

UPPSALA UNIVERSITET

- In a given state we want to maximize future rewards $R_{t+1}, R_{t+2}, \ldots$

The return

$$G_t = R_{t+1} + R_{t+2} + R_{t+3} + \cdots = \sum_{k=0}^{\infty} R_{t+k+1}$$

- In a given state we want to maximize future rewards $R_{t+1}, R_{t+2}, \ldots$

The discounted return

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}, \text{ where } 0 < \gamma \leq 1.$$

The return

- In a given state we want to maximize future rewards $R_{t+1}, R_{t+2}, \ldots$

The discounted return

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}, \text{ where } 0 < \gamma \leq 1.$$

The discount rate γ:

- If $\gamma < 1$, we put less value on future rewards.

The return

- In a given state we want to maximize future rewards $R_{t+1}, R_{t+2}, \ldots$

The discounted return

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}, \text{ where } 0 < \gamma \leq 1.$$

The discount rate γ:

- If $\gamma < 1$, we put less value on future rewards.
- If $\gamma < 1$, G_t will be finite as long as R_k are bounded!

- In a given state we want to maximize future rewards $R_{t+1}, R_{t+2}, \ldots$

The discounted return

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}, \text{ where } 0 < \gamma \leq 1.$$

The discount rate γ:

- If $\gamma < 1$, we put less value on future rewards.
- If $\gamma < 1$, G_t will be finite as long as R_k are bounded!
- It is sometimes possible to use undiscounted returns ($\gamma = 1$), e.g., if the task always ends after a finite number of steps.

Value Functions

The state-value function

UPPSALA UNIVERSITET

- Note that S_t and R_t etc are random variables.

The state-value function

- Note that S_t and R_t etc are random variables.
- Hence the return is also a random variable:

$$G_t = R_{t+1} + \gamma R_{t+2} + \cdots$$

The state-value function

- Note that S_t and R_t etc are random variables.
- Hence the return is also a random variable:

$$G_t = R_{t+1} + \gamma R_{t+2} + \cdots$$

- We thus consider the *expected* return.

- Note that S_t and R_t etc are random variables.
- Hence the return is also a random variable:

$$G_t = R_{t+1} + \gamma R_{t+2} + \cdots$$

- We thus consider the *expected* return.

The state-value function

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

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

- Note that S_t and R_t etc are random variables.
- Hence the return is also a random variable:

$$G_t = R_{t+1} + \gamma R_{t+2} + \cdots$$

- We thus consider the *expected* return.

The state-value function

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

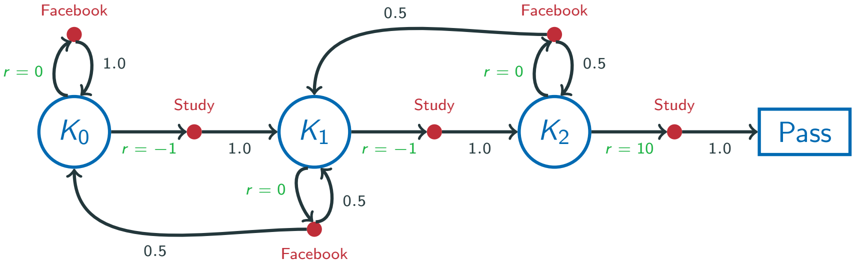
$$v_\pi(s) = \mathbb{E}_\pi[G_t | S_t = s].$$

- **Prediction:** Compute $v_\pi(s)$.

UPPSALA UNIVERSITET

Discount: $\gamma = 0.9$.

Policy: $\pi(a|s) = 0.5$ for all a and s.

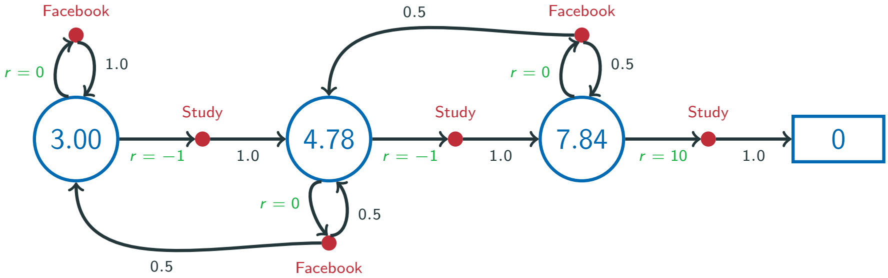


Example

Discount: $\gamma = 0.9$.

Policy: $\pi(a|s) = 0.5$ for all a and s.

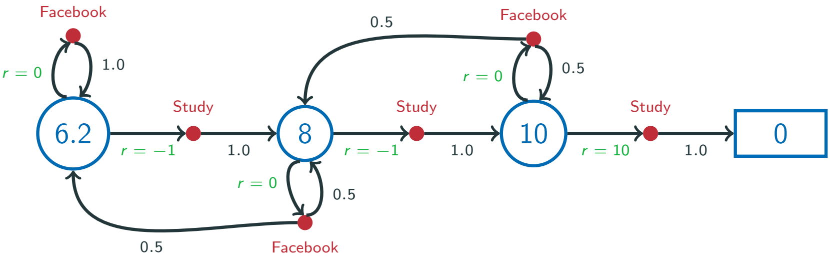
$v_\pi(s)$



Example

Discount: $\gamma = 0.9$.

Policy: Always choose study.



The action-value function

- Another important value function is the *action-value function*.

The action-value function

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

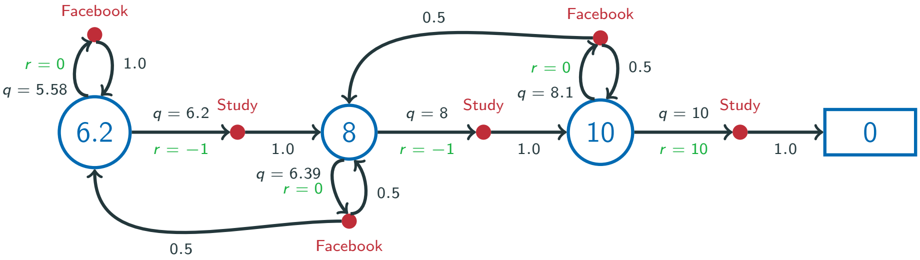
$$q_\pi(s, a) = \mathbb{E}_\pi[G_t | S_t = s, A_t = a].$$

- Often called the Q-function.

Discount: $\gamma = 0.9$.

Policy: Always choose study.

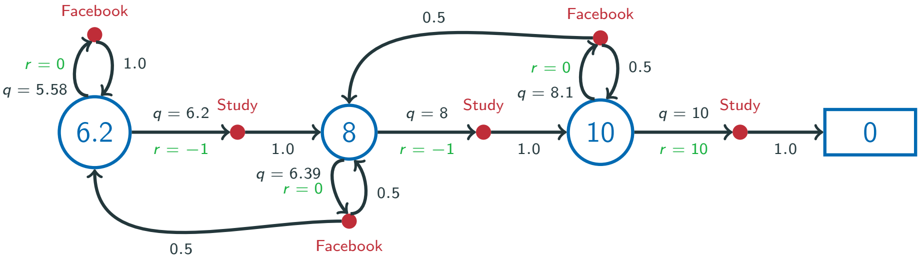
Action-values, q.



Discount: $\gamma = 0.9$.

Policy: Always choose study.

Action-values, q.



"If I just this one time choose Facebook, and after that follow the policy (always Study), what will my expected discounted return be?"

Bellman equations

- Return: Note that

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots =$$

UPPSALA UNIVERSITET

- Return: Note that

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots = R_{t+1} + \gamma G_{t+1}.$$

The Bellman Equation

- Return: Note that

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots = R_{t+1} + \gamma G_{t+1}.$$

- Hence, the value function satisfies to following equation:

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

- Return: Note that

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots = R_{t+1} + \gamma G_{t+1}.$$

- Hence, the value function satisfies to following equation:

$$\begin{aligned} v_\pi(s) &= \mathbb{E}_\pi[G_t | S_t = s] \\ &= \mathbb{E}_\pi[R_{t+1} + \gamma G_{t+1} | S_t = s] \end{aligned}$$

- Return: Note that

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots = R_{t+1} + \gamma G_{t+1}.$$

- Hence, the value function satisfies to following equation:

$$\begin{aligned} v_\pi(s) &= \mathbb{E}_\pi[G_t | S_t = s] \\ &= \mathbb{E}_\pi[R_{t+1} + \gamma G_{t+1} | S_t = s] \\ &= \mathbb{E}_\pi[R_{t+1} + \gamma v_\pi(S_{t+1}) | S_t = s] \end{aligned}$$

- Return: Note that

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots = R_{t+1} + \gamma G_{t+1}.$$

- Hence, the value function satisfies to following equation:

$$\begin{aligned} v_\pi(s) &= \mathbb{E}_\pi[G_t | S_t = s] \\ &= \mathbb{E}_\pi[R_{t+1} + \gamma G_{t+1} | S_t = s] \\ &= \mathbb{E}_\pi[R_{t+1} + \gamma v_\pi(S_{t+1}) | S_t = s] \end{aligned}$$

- "The value of s is the expected immediate reward plus the discounted expected value of the next state".

- Return: Note that

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots = R_{t+1} + \gamma G_{t+1}.$$

- Hence, the value function satisfies to following equation:

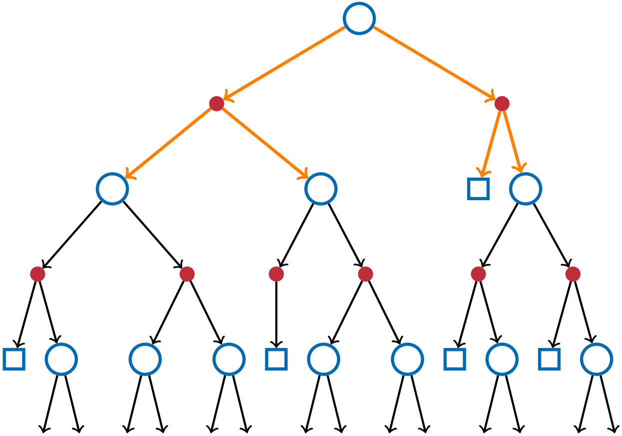
$$\begin{aligned} v_\pi(s) &= \mathbb{E}_\pi[G_t | S_t = s] \\ &= \mathbb{E}_\pi[R_{t+1} + \gamma G_{t+1} | S_t = s] \\ &= \mathbb{E}_\pi[R_{t+1} + \gamma v_\pi(S_{t+1}) | S_t = s] \end{aligned}$$

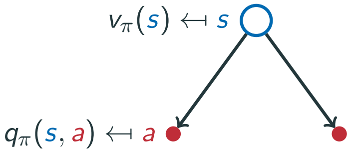
- "The value of s is the expected immediate reward plus the discounted expected value of the next state".
- In the same way

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

UPPSALA UNIVERSITET

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

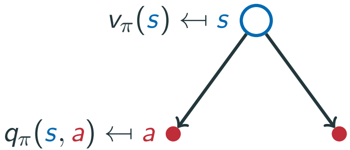




- The state-value of s is the expected action-value:

$$v_\pi(s) = \sum_a \pi(a|s) q_\pi(s, a).$$

The expectation

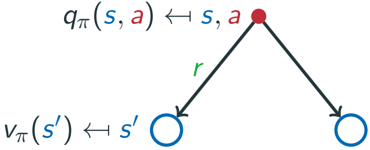


- The state-value of s is the expected action-value:

$$v_\pi(s) = \sum_a \pi(a|s) q_\pi(s, a).$$

- For a deterministic policy $a = \pi(s)$ we get $v_\pi(s) = q_\pi(s, \pi(s))$.

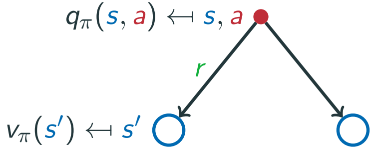
The expectation



- Given s and a, the immediate reward r and the next state s' has prob $p(s', r|s, a)$. So,

$$q_\pi(s, a) =$$

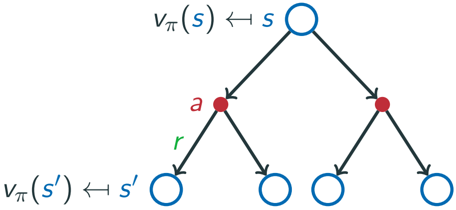
The expectation



- Given s and a, the immediate reward r and the next state s' has prob $p(s', r|s, a)$. So,

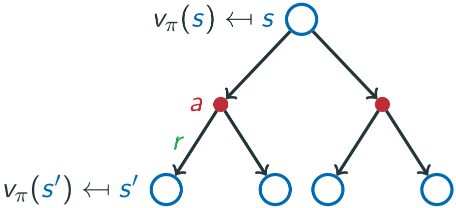
$$q_\pi(s, a) = \sum_{r, s'} p(s', r|s, a)(r + \gamma v_\pi(s')).$$

The Bellman equation for v_π



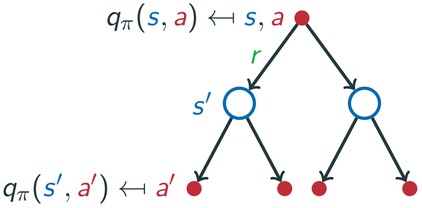
$$v_\pi(s) = \sum_a \pi(a|s) q_\pi(s, a)$$

The Bellman equation for v_π



$$\begin{aligned} v_\pi(s) &= \sum_a \pi(a|s) q_\pi(s, a) \\ &= \sum_a \pi(a|s) \sum_{r,s'} p(s', r|s, a)[r + \gamma v_\pi(s')]. \end{aligned}$$

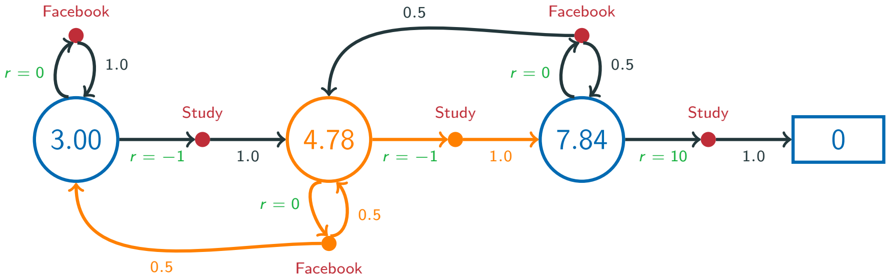
The Bellman equation for q_π



$$q_\pi(s, a) = \sum_{r, s'} p(s', r | s, a) \left[r + \gamma \sum_{a'} \pi(a' | s') q_\pi(s', a') \right]$$

Discount: $\gamma = 0.9$.

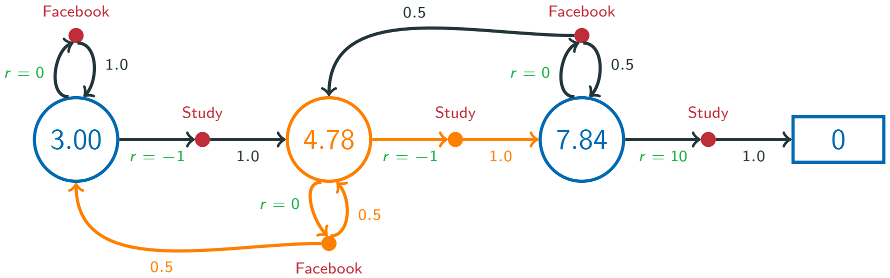
Policy: $\pi(a|s) = 0.5$ for all a and s.



$$v_\pi(s) = \sum_a \pi(a|s) q_\pi(s, a), \quad q_\pi(s, a) = \sum_{r, s'} p(s', r | s, a)[r + \gamma v_\pi(s')].$$

Discount: $\gamma = 0.9$.

Policy: $\pi(a|s) = 0.5$ for all a and s.



$$v_\pi(s) = \sum_a \pi(a|s) q_\pi(s,a), \quad q_\pi(s,a) = \sum_{r,s'} p(s', r|s,a)[r + \gamma v_\pi(s')].$$

$$4.78 = \underbrace{0.5}_{\pi(\text{facebook}|s)} \times \underbrace{\gamma[0.5 \times 3.00 + 0.5 \times 4.78]}_{q_\pi(s,\text{facebook})} + \underbrace{0.5}_{\pi(\text{study}|s)} \times \underbrace{[-1 + \gamma \times 7.84]}_{q_\pi(s,\text{study})}$$

Solving the Bellman equation

$$v_\pi(s) = \sum_a \pi(a|s) \sum_{r,s'} p(s', r|s, a)[r + \gamma v_\pi(s')].$$

- A system of linear equations in $v_\pi(s)$.

Solving the Bellman equation

UPPSALA UNIVERSITET

$$v_\pi(s) = \sum_a \pi(a|s) \sum_{r,s'} p(s', r|s, a)[r + \gamma v_\pi(s')].$$

- A system of linear equations in $v_\pi(s)$.
- One equation for each $s \in \mathcal{S}$.

Solving the Bellman equation

$$v_\pi(s) = \sum_a \pi(a|s) \sum_{r,s'} p(s', r|s, a)[r + \gamma v_\pi(s')].$$

- A system of linear equations in $v_\pi(s)$.
- One equation for each $s \in \mathcal{S}$.
- A unique solution, that can be expressed analytically.

Solving the Bellman equation

UPPSALA UNIVERSITET

$$v_\pi(s) = \sum_a \pi(a|s) \sum_{r,s'} p(s', r|s, a)[r + \gamma v_\pi(s')].$$

- A system of linear equations in $v_\pi(s)$.
- One equation for each $s \in \mathcal{S}$.
- A unique solution, that can be expressed analytically.
- If $\mathcal{S}$ is large, more efficient to use iterative solutions (Lecture 3).

Solving the Bellman equation

$$v_\pi(s) = \sum_a \pi(a|s) \sum_{r,s'} p(s', r|s, a)[r + \gamma v_\pi(s')].$$

- A system of linear equations in $v_\pi(s)$.
- One equation for each $s \in \mathcal{S}$.
- A unique solution, that can be expressed analytically.
- If $\mathcal{S}$ is large, more efficient to use iterative solutions (Lecture 3).
- If $p(s', r|s, a)$ is not known, we have to learn $v_\pi(s)$ from experience (Lecture 4).

Solving the Bellman equation

$$v_\pi(s) = \sum_a \pi(a|s) \sum_{r,s'} p(s', r|s, a)[r + \gamma v_\pi(s')].$$

- A system of linear equations in $v_\pi(s)$.
- One equation for each $s \in \mathcal{S}$.
- A unique solution, that can be expressed analytically.
- If $\mathcal{S}$ is large, more efficient to use iterative solutions (Lecture 3).
- If $p(s', r|s, a)$ is not known, we have to learn $v_\pi(s)$ from experience (Lecture 4).
- If $\mathcal{S}$ is infinite, we can't compute the value for each state individually, and instead have find some function $\hat{v}(s, \boldsymbol{w}) \approx v_\pi(s)$. (Second part of course)

Optimal Value Functions

UPPSALA UNIVERSITET

- **Optimal state-value function:**

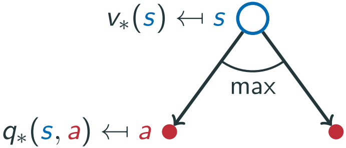
$$v_*(s) = \max_\pi v_\pi(s), \quad \text{for all } s \in \mathcal{S}.$$

- **Optimal action-value function:**

$$q_*(s,a) = \max_\pi q_\pi(s,a), \quad \text{for all } s \in \mathcal{S} \text{ and } a \in \mathcal{A}.$$

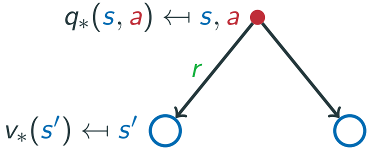
Bellman equation for v_*

The optimal $v_*(s)$ should be the maximum of $q_*(s, a)$.



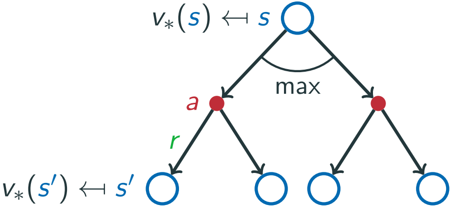
$$v_*(s) = \max_a q_*(s, a).$$

Bellman equation for q_*

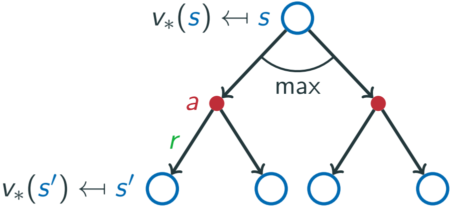


$$q_*(s, a) = \sum_{r, s'} p(s', r | s, a)(r + \gamma v_*(s')).$$

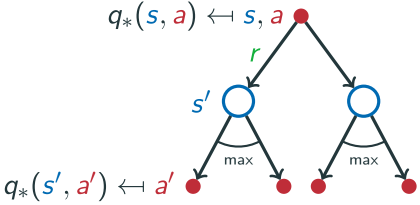
The Bellman equation for v_*



$$v_*(s) = \max_a \sum_{r,s'} p(s', r | s, a)[r + \gamma v_*(s')]$$

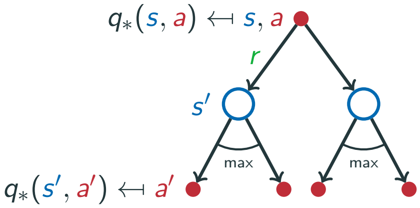


$$
\begin{aligned}
v_*(s) &= \max_a \sum_{r,s'} p(s', r|s, a)[r + \gamma v_*(s')] \\
&= \max_a \mathbb{E}[R_{t+1} + \gamma v_*(S_{t+1})|S_t = s, A_t = a]
\end{aligned}
$$



$$q_*(s, a) = \sum_{r, s'} p(s', r | s, a) \left[r + \gamma \max_{a'} q_*(s', a') \right]$$

The Bellman equation for q_*



$$q_*(s,a) = \sum_{r,s'} p(s', r | s, a) \left[r + \gamma \max_{a'} q_*(s', a') \right]$$

$$= \mathbb{E}[R_{t+1} + \gamma \max_{a'} q_*(S_{t+1}, a') | S_t = s, A_t = a]$$

Solving the Bellman optimality equation

$$v_*(s) = \max_a \sum_{r,s'} p(s', r | s, a)[r + \gamma v_*(s')]$$

- A system of *non*-linear equations.

Solving the Bellman optimality equation

$$v_*(s) = \max_a \sum_{r,s'} p(s', r|s, a)[r + \gamma v_*(s')]$$

- A system of *non*-linear equations.
- One equation for each s.

Solving the Bellman optimality equation

$$v_*(s) = \max_a \sum_{r,s'} p(s', r|s, a)[r + \gamma v_*(s')]$$

- A system of *non*-linear equations.
- One equation for each s.
- In general no closed-form solution!

Solving the Bellman optimality equation

$$v_*(s) = \max_a \sum_{r,s'} p(s', r|s, a)[r + \gamma v_*(s')]$$

- A system of *non*-linear equations.
- One equation for each s.
- In general no closed-form solution!
- But there are iterative solution methods (Lecture 3).

Optimal Policy

Optimal policy

Partial ordering over policies:

$$\pi \geq \pi' \text{ if } v_\pi(s) \geq v_{\pi'}(s), \text{ for all } s.$$

Optimal policy

UPPSALA UNIVERSITET

Partial ordering over policies:

$$\pi \geq \pi' \text{ if } v_\pi(s) \geq v_{\pi'}(s), \text{ for all } s.$$

Theorem

- There exists (at least one) optimal policy π_* such that $\pi_* \geq \pi$ for all policies π.

Partial ordering over policies:

$$\pi \geq \pi' \text{ if } v_\pi(s) \geq v_{\pi'}(s), \text{ for all } s.$$

Theorem

- There exists (at least one) optimal policy π_* such that $\pi_* \geq \pi$ for all policies π.
- All optimal policies achieve the optimal state-value function $v_{\pi_*}(s) = v_*(s)$.

Optimal policy

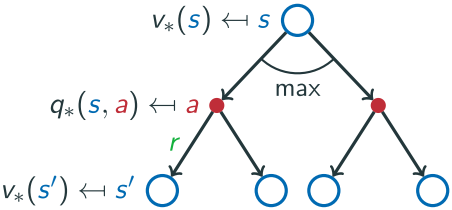
Partial ordering over policies:

$$\pi \geq \pi' \text{ if } v_\pi(s) \geq v_{\pi'}(s), \text{ for all } s.$$

Theorem

- There exists (at least one) optimal policy π_* such that $\pi_* \geq \pi$ for all policies π.
- All optimal policies achieve the optimal state-value function $v_{\pi_*}(s) = v_*(s)$.
- All optimal policies achieve the optimal action-value function $q_{\pi_*}(s, a) = q_*(s, a)$.

Optimal policy



What to do in state s?

1. Choose an a that maximize the optimal action-value $q_*(s, a)$.
2. Then use an optimal policy from s'.

Finding an optimal policy

- **Control:** Find optimal policy.

- **Control:** Find optimal policy.
- The policy

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

is optimal.

- **Control:** Find optimal policy.
- The policy

$$\pi_*(s) = \arg\max_{a} q_*(s, a) = \arg\max_{a} \sum_{r, s'} p(s', r|s, a)[r + \gamma v_*(s')]$$

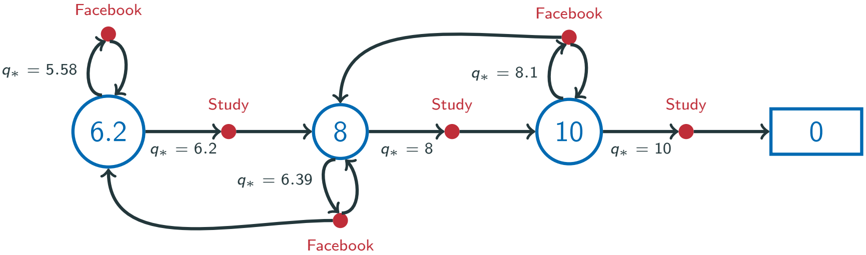
is optimal.

- **Control:** Find optimal policy.
- The policy

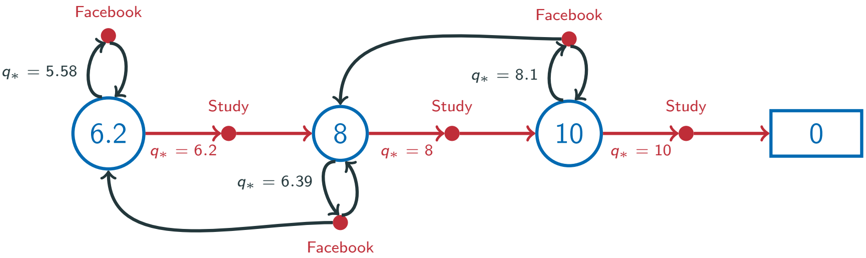
$$\pi_*(s) = \arg\max_a q_*(s, a) = \arg\max_a \sum_{r,s'} p(s', r|s, a)[r + \gamma v_*(s')]$$

  is optimal.
- **Note:** If we know $q_*(s, a)$ we don't need the dynamics to find an optimal policy!

Example: Optimal state-value and action-value

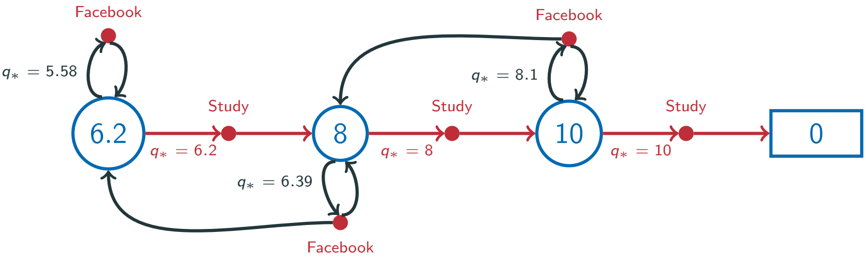


Example: Optimal policy



Optimal policy: Always study!

Example: Optimal policy



Optimal policy: Always study!

...according to this MDP. In real life it may be good to take a break once in a while.

Summary

- Markov Decision Processes
- Discounted return
- Value functions
- Bellman equations

Summary

- Markov Decision Processes
- Discounted return
- Value functions
- Bellman equations

Next:

- Find value functions and optimal policy if $p(s', r|s, a)$ is known. (Lecture 3)
- What if $p(s', r|s, a)$ is not known? (Lecture 4-5)
- Tinkering Notebook 2: You can do Section 1-5 now.
- Assignment 1: You can look at Problem 1 – Problem 3a.