Markov Decision Processes

Corrado Possieri

Machine and Reinforcement Learning in Control Applications

Introduction

- **Markov decision processes** (MDP) formally describe an environment for reinforcement learning.
- The environment is fully observable
  - the current state completely characterizes the future.
- Almost all learning problems can be formalized as MDPs:
  - optimal control deals with continuous MDPs;
  - partially observable problems can be converted into MDPs;
  - bandits are MDPs with just a single state.

Markov property

Markov property

The future is independent of the past given the present.

Formally

$$\mathbb{P}[S_{t+1}|S_t] = \mathbb{P}[S_{t+1}|S_1, S_2, \ldots, S_t].$$

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

State transition matrix

- Given states s and s', the state transition probability is

$$P_{s,s'} = \mathbb{P}[S_{t+1} = s' | S_t = s].$$

- If the states are **finite**
  - define the state transition matrix

$$P = \begin{array}{cc} & \text{to} \\ \text{from} & \begin{bmatrix} P_{1,1} & P_{1,2} & \cdots & P_{1,n} \\ \vdots & \vdots & \ddots & \vdots \\ P_{n,1} & P_{n,2} & \cdots & P_{n,n} \end{bmatrix} \end{array};$$

  - each row of P sums to 1:

$$\sum_{j=1}^{n} P_{i,j} = 1, \quad i = 1, \ldots, n.$$

Probability distributions in Markov chains

- Letting

$$\pi(t) = \begin{bmatrix} \mathbb{P}[S_t = 1] \\ \mathbb{P}[S_t = 2] \\ \vdots \\ \mathbb{P}[S_t = n] \end{bmatrix}^\top,$$

one has

$$\pi(t+1) = \pi(t)\,P.$$

- **Stationary distributions** satisfy

$$\bar{\pi} = \bar{\pi}\,P.$$

- We have that

$$\mathbb{P}[S_{t+h} = s' | S_t = s] = [P^h]_{ij}.$$

Markov process

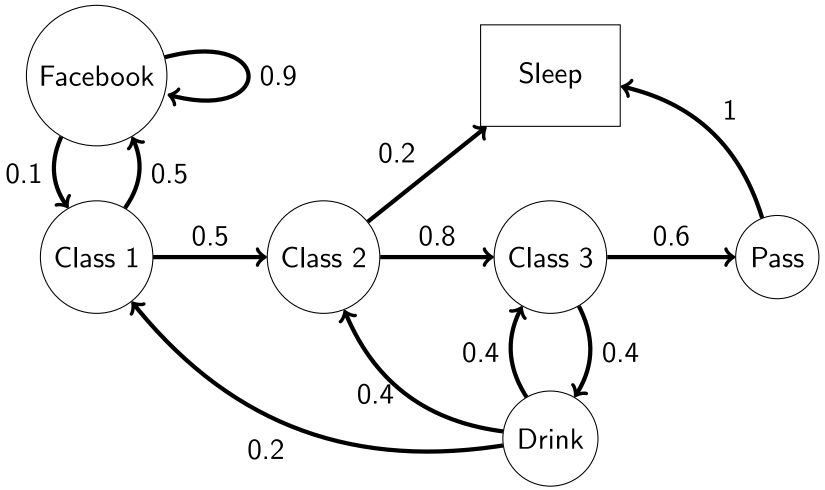
A **Markov process** is a memoryless random process, *i.e.*, a sequence of random states $S_1, S_2, \ldots$ with the Markov property.

Markov chain

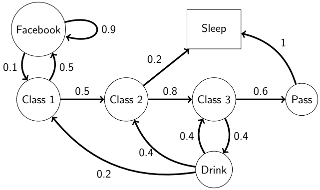
A *Markov chain* is a pair $(\mathcal{S}, P)$ with

1. $\mathcal{S}$ is a finite set of states;
2. P is a the transition matrix.

Student Markov chain



Student Markov chain episodes



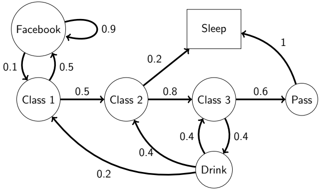
Sample **episodes**

$$S_1, S_2, \ldots, S_T,$$

starting with $S_1 =$ C1:

- C1 C2 C3 P S;
- C1 F F F C1 C2 S;
- C1 C2 C3 D C1 C2 C3 D C2 S;
- C1 C2 C3 D C1 F F F F C1 C2 C3 D C2 C3 D C3 P.

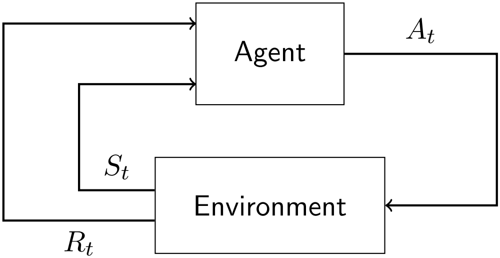
Student Markov chain transition matrix



$$
\begin{array}{cc}
 & \begin{array}{ccccccc} \mathsf{C1} & \mathsf{C2} & \mathsf{C3} & \mathsf{P} & \mathsf{D} & \mathsf{F} & \mathsf{S} \end{array} \\
\begin{array}{c} \mathsf{C1} \\ \mathsf{C2} \\ \mathsf{C3} \\ \mathsf{P} \\ \mathsf{D} \\ \mathsf{F} \\ \mathsf{S} \end{array} &
\begin{bmatrix}
 & 0.5 & & & & 0.5 & \\
 & & 0.8 & & & & 0.2 \\
 & & & 0.6 & 0.4 & & \\
 & & & & & & 1 \\
0.2 & 0.4 & 0.4 & & & & \\
0.1 & & & & & 0.9 & \\
 & & & & & & 1
\end{bmatrix}
\end{array}
$$

Agent and environment

- The **agent** is the decision maker.
- The **environment** is everything outside the agent.
- These interact continually
  - the agent takes actions;
  - the environment presents new situations and gives rewards.



Interactions between agent and environment

- At each time step
  - the agent observes the environment's *state* $S_t \in \mathcal{S}$;
  - the agent selects an *action* $A_t \in \mathcal{A}(S_t)$;
  - the agent receives the *reward* $R_{t+1} \in \mathcal{R}$;
  - the agent finds itself in the new state $S_{t+1} \in \mathcal{S}$.

- Therefore a trajectory of an MDP is

$$S_0, A_0, R_1, S_1, A_1, R_2, S_2, \ldots.$$

- The *dynamics* of an MDP are defined by

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

with

$$\sum_{s' \in \mathcal{S}} \sum_{r \in \mathcal{R}} p(s', r|s, a) = 1, \quad \forall a \in \mathcal{A}(s), \forall s \in \mathcal{S}.$$

Markov property in MDPs

- The state s of an MDP satisfies the Markov property.
- $\mathbb{P}[S_{t+1}, R_{t+1}]$ depends only on S_t and A_t.
- This is an assumption about the representation
  - not the process.
- Markov state can be learned from non-Markov observations.

Some probability functions

- From $p(s', r|s, a)$ we can define other probability functions:
  - state-transition probabilities

$$\begin{aligned} p(s'|s, a) &= \mathbb{P}[S_{t+1} = s'|S_t = s, A_t = a] \\ &= \sum_{r \in \mathcal{R}} p(s', r|s, a); \end{aligned}$$

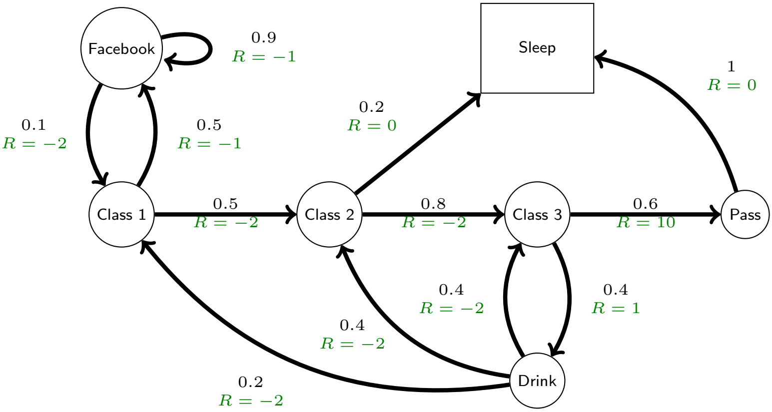
  - expected rewards for state–action pairs

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

  - expected rewards for state–action–next action triples

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

Student Markov chain with rewards



The reward hypothesis

Reward hypothesis

That all of what we mean by goals and purposes can be well thought of as the maximization of the expected value of the cumulative sum of a received scalar signal (called reward).

Some examples of rewards:

escape from a maze: -1 at each time step in the maze;

completing task: 1 at each step at which the task is completed and 0 otherwise

checkers: $+1$ for winning and -1 for losing a game.

Returns

Episodic tasks

- If there is a natural notion of final time step
  - the agent–environment interaction breaks naturally into subsequences, which we call **episodes**;
  - the time of termination T is a random variable that varies from episode to episode;
  - each episode ends in a special state called the **terminal state**, followed by a reset;
  - we use $\mathcal{S}^+$ to denote $\mathcal{S}$ and the terminal states;
  - the **expected return** is the sum of rewards

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

Returns

Continuing tasks

- If we are dealing with continuing task
  - introduce a **discount factor** $\gamma \in [0, 1]$;
  - the **expected return** is the sum of discounted rewards

$$
\begin{aligned}
G_t &= R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots \\
&= \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}.
\end{aligned}
$$

  - delayed rewards are discounted

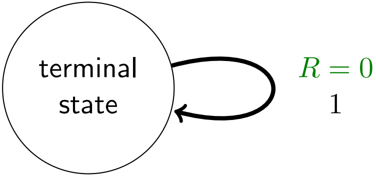
$$G_t = R_{t+1} + \gamma G_{t+1};$$

  - this values immediate reward above delayed reward
    - $\gamma \to 0$ leads to greedy evaluation;
    - $\gamma \to 1$ leads to far-sighted evaluation;
- discounting with $\gamma < 1$ avoids infinite returns if $\mathcal{R}$ is bounded.

Returns

Unifying notation

- The terminal state of an episodic task can be thought as an absorbing state generating reward 0.



- With such a convention, the return can be defined as

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

even for episodic tasks.

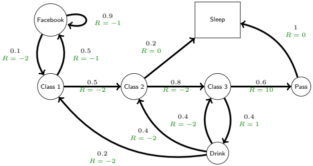
Policy

- A policy is a mapping from the current state $s \in \mathcal{S}$ to probabilities of selecting actions $a \in \mathcal{A}(s)$.
- If the agent is following policy π

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

- Reinforcement learning methods specify how the agent's policy is changed as a result of its experience.

Student Markov chain policy



Policy:

$$\begin{aligned}
&\pi(\mathsf{C2}|\mathsf{C1}) = 0.5, \quad \pi(\mathsf{F}|\mathsf{C1}) = 0.5,\\
&\pi(\mathsf{C3}|\mathsf{C2}) = 0.8, \quad \pi(\mathsf{S}|\mathsf{C2}) = 0.2,\\
&\pi(\mathsf{P}|\mathsf{C3}) = 0.6, \quad \pi(\mathsf{D}|\mathsf{C3}) = 0.4,\\
&\pi(\mathsf{S}|\mathsf{P}) = 1,\\
&\pi(\mathsf{C1}|\mathsf{D}) = 0.2, \quad \pi(\mathsf{C2}|\mathsf{D}) = 0.4, \quad \pi(\mathsf{C3}|\mathsf{D}) = 0.4,\\
&\pi(\mathsf{C1}|\mathsf{F}) = 0.1, \quad \pi(\mathsf{F}|\mathsf{F}) = 0.9,\\
&\pi(\mathsf{S}|\mathsf{S}) = 1.
\end{aligned}$$

Value function

Value functions estimate how good it is for the agent to be in a given state (or state–action pairs).

State value function

The value function of a state s under a policy π is the expected return when starting in s and following π thereafter:

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

State-action value function

The value function of a state s and of action a under a policy π is the expected return starting from s, taking the action a, and following policy π thereafter:

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

Bellman equation

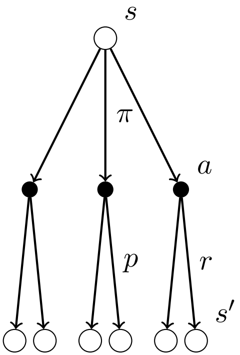
- We can obtain a consistency condition for v_π:

$$\begin{aligned} v_\pi(s) &= \mathbb{E}_\pi[G_t | S_t = s] \\ &= \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | S_t = s\right] \\ &= \mathbb{E}_\pi\left[R_{t+1} + \gamma G_{t+1} | S_t = s\right] \\ &= \sum_{a\in\mathcal{A}(s)} \pi(a|s) \sum_{s'\in\mathcal{S}^+, r\in\mathcal{R}} p(s', r|s, a)\left(r + \gamma \mathbb{E}_\pi[G_{t+1}|S_{t+1} = s']\right) \\ &= \sum_{a\in\mathcal{A}(s)} \pi(a|s) \sum_{s'\in\mathcal{S}^+, r\in\mathcal{R}} p(s', r|s, a)\left(r + \gamma v_\pi(s')\right). \end{aligned}$$

- For each triple r, a, s'
  - compute its probability $\pi(a, s)p(s', r|s, a)$;
  - compute the expected return $r + \gamma v_\pi(s')$.
- Sum over all possibilities to get an expected value.

Backup diagram of the Bellman equation

$$v_\pi(s) = \sum_{a \in \mathcal{A}(s)} \pi(a|s) \sum_{s' \in \mathcal{S}^+, r \in \mathcal{R}} p(s', r|s, a) \left(r + \gamma v_\pi(s')\right).$$



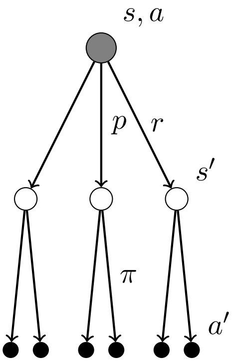
Relation between v_π and q_π

- It can be easily derived that
  - $v_\pi(s) = \sum_{a \in \mathcal{A}(s)} \pi(a|s) q_\pi(s,a)$;
  - $q_\pi(s,a) = \sum_{s' \in \mathcal{S}, r \in \mathcal{R}} p(s', r|s,a) \left(r + \gamma v_\pi(s')\right)$.
- This allows us to derive a Bellman equation for q_π:

$$q_\pi(s,a) = \sum_{s' \in \mathcal{S}, r \in \mathcal{R}} p(s', r|s,a) \left(r + \gamma \sum_{a' \in \mathcal{A}(s')} \pi(a'|s') q_\pi(s', a') \right).$$

Backup diagram of the Bellman equation for q_π

$$q_\pi(s,a) = \sum_{s' \in \mathcal{S}, r \in \mathcal{R}} p(s', r | s, a) \left(r + \gamma \sum_{a' \in \mathcal{A}(s')} \pi(a'|s') q_\pi(s', a') \right).$$



Value function for the student Markov chain

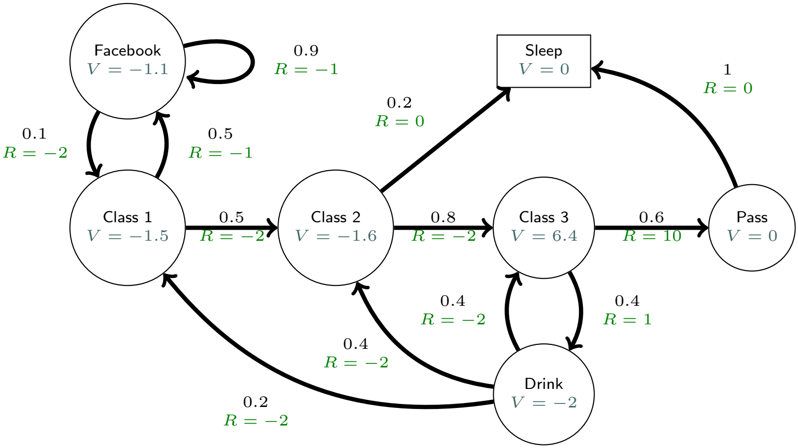
Bellman equation

The Bellman equation for the student Markov chain reads as

$$\begin{aligned}
v_\pi(\mathsf{C1}) &= 0.5(\gamma v_\pi(\mathsf{C2}) - 2) + 0.5(\gamma v_\pi(\mathsf{F}) - 1),\\
v_\pi(\mathsf{C2}) &= 0.8(\gamma v_\pi(\mathsf{C3}) - 2) + 0.2\gamma v_\pi(\mathsf{S}),\\
v_\pi(\mathsf{C3}) &= 0.4(\gamma v_\pi(\mathsf{D}) + 1) + 0.6(\gamma v_\pi(\mathsf{P}) + 10),\\
v_\pi(\mathsf{P}) &= \gamma v_\pi(\mathsf{S}),\\
v_\pi(\mathsf{D}) &= 0.2(\gamma v_\pi(\mathsf{C1}) - 2) + 0.4(\gamma v_\pi(\mathsf{C2}) - 2),\\
&\qquad + 0.4(\gamma v_\pi(\mathsf{C3}) - 2),\\
v_\pi(\mathsf{F}) &= 0.1(\gamma v_\pi(\mathsf{C1}) - 2) + 0.9(\gamma v_\pi(\mathsf{F}) - 1),\\
v_\pi(\mathsf{S}) &= \gamma v_\pi(\mathsf{S}).
\end{aligned}$$

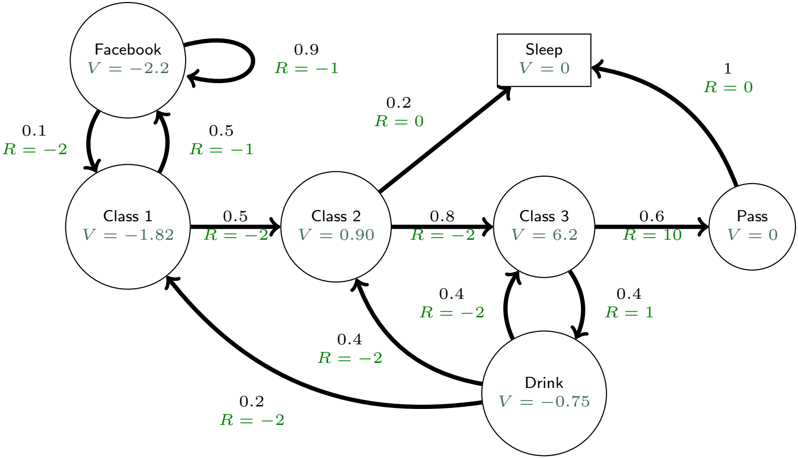
Value function for the student Markov chain

$\gamma = 0$



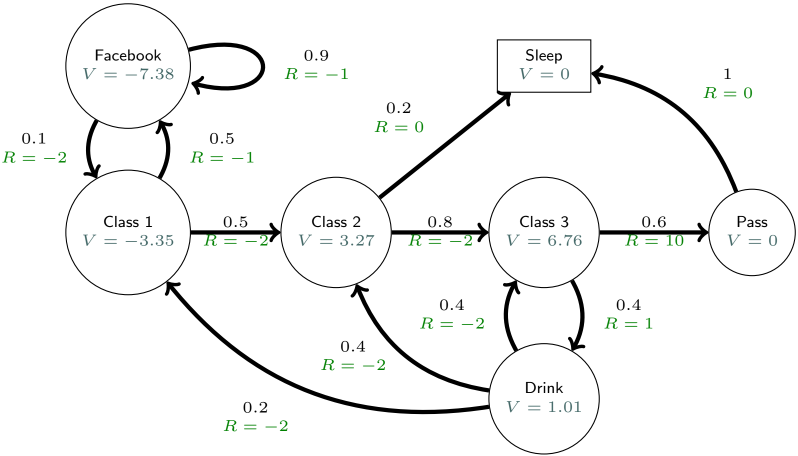
Value function for the student Markov chain

$\gamma = 0.5$



Value function for the student Markov chain

$\gamma = 0.9$



Optimal value function

- Value functions can be used to sort policies
  - $\pi \geqslant \pi' \iff v_\pi(s) \geqslant v_{\pi'}(s), \forall s \in \mathcal{S}.$
- There is always at least one policy that is better than or equal to all other policies
  - this is an optimal policy, denoted π_*;
  - all policies π_* share the same value function
    - this is the **optimal value function**

$$v_\star(s) = \max_\pi v_\pi(s);$$

- all policies π_* share the same optimal action-value function

$$\begin{aligned} q_\star(s,a) &= \max_\pi q_\pi(s,a) \\ &= \mathbb{E}[R_{t+1} + \gamma v_\star(S_{t+1}) | S_t = s, A_t = a]. \end{aligned}$$

Bellman optimality equation

- v_{π_*} and $q_\pi(*)$ must satisfy the Bellman equation.
- Further, it must hold that

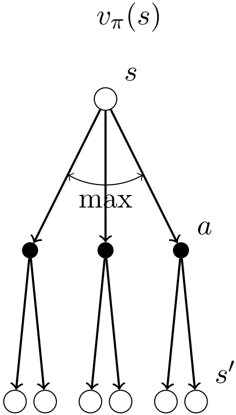
$$v_*(s) = \max_{a \in \mathcal{A}(s)} q_*(s,a).$$

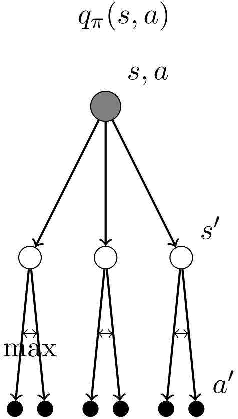
- We thus obtain the **Bellman optimality equation**

$$v_*(s) = \max_{a \in \mathcal{A}(s)} \sum_{s' \in \mathcal{S}^+, r \in \mathcal{R}} p(s', r | s, a) \left(r + \gamma v_*(s') \right),$$

$$q_*(s,a) = \sum_{s' \in \mathcal{S}, r \in \mathcal{R}} p(s', r | s, a) \left(r + \gamma \max_{a' \in \mathcal{A}(s')} q_\pi(s', a') \right).$$

Backup diagrams of the Bellman optimality equation





Considerations on the Bellman optimality equation

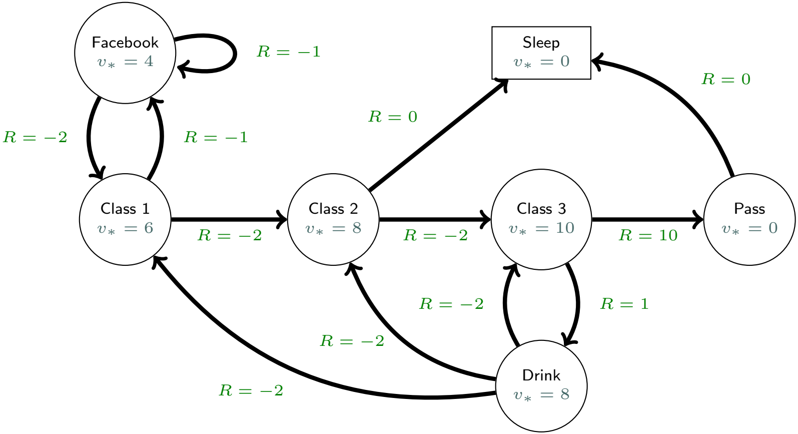
- The Bellman optimality equation
  - is nonlinear;
  - no closed form solution exists;
  - can be solved explicitly in some cases
    - Dijkstra's algorithm;
    - A* search algorithm;
- Optimal actions at state s can be determined as

$$a \leftarrow \max_a q_*(s, a)$$

  - there is always a deterministic optimal policy for any MDP.

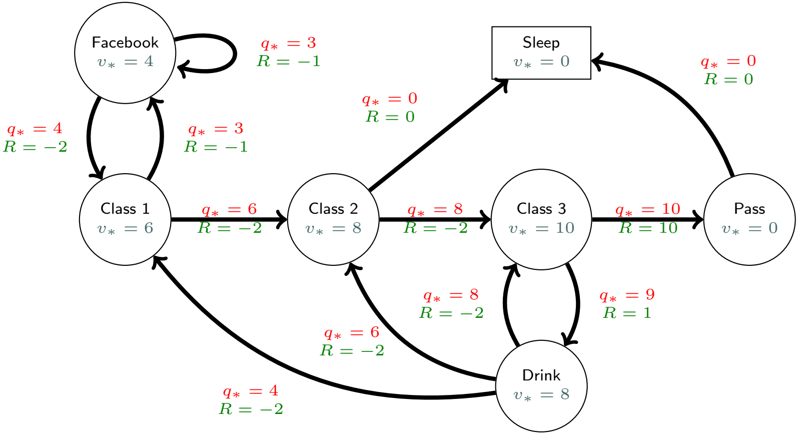
Optimal value function in student MDP

γ=1



Optimal state-action value function in student MDP

γ=1



Issues on solving the Bellman optimality equation

- The dynamics of the environment are not accurately known;
- Computationally expensive;
- The states may not have the Markov property.
- Many iterative solution methods
  - value iteration;
  - policy iteration;
  - q-learning;
  - SARSA.