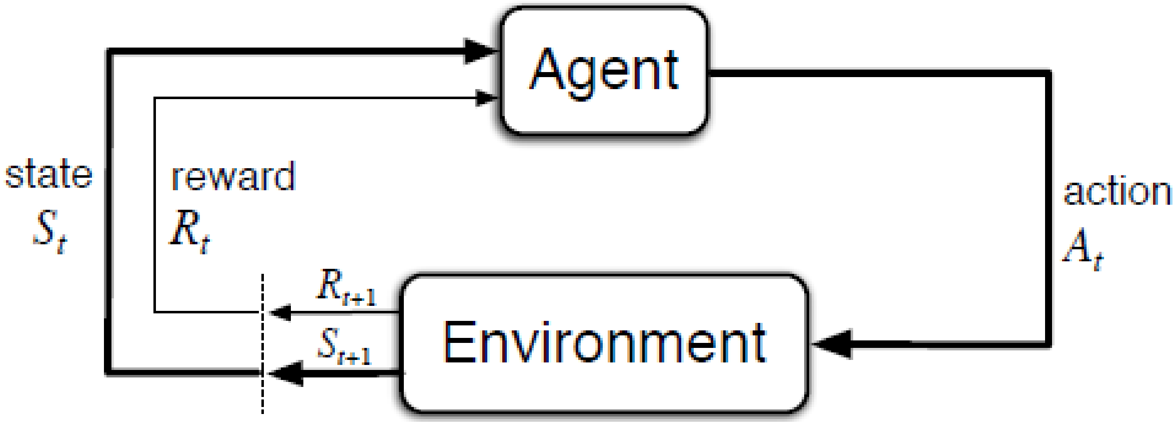
Markov Decision Processes (MDPs)

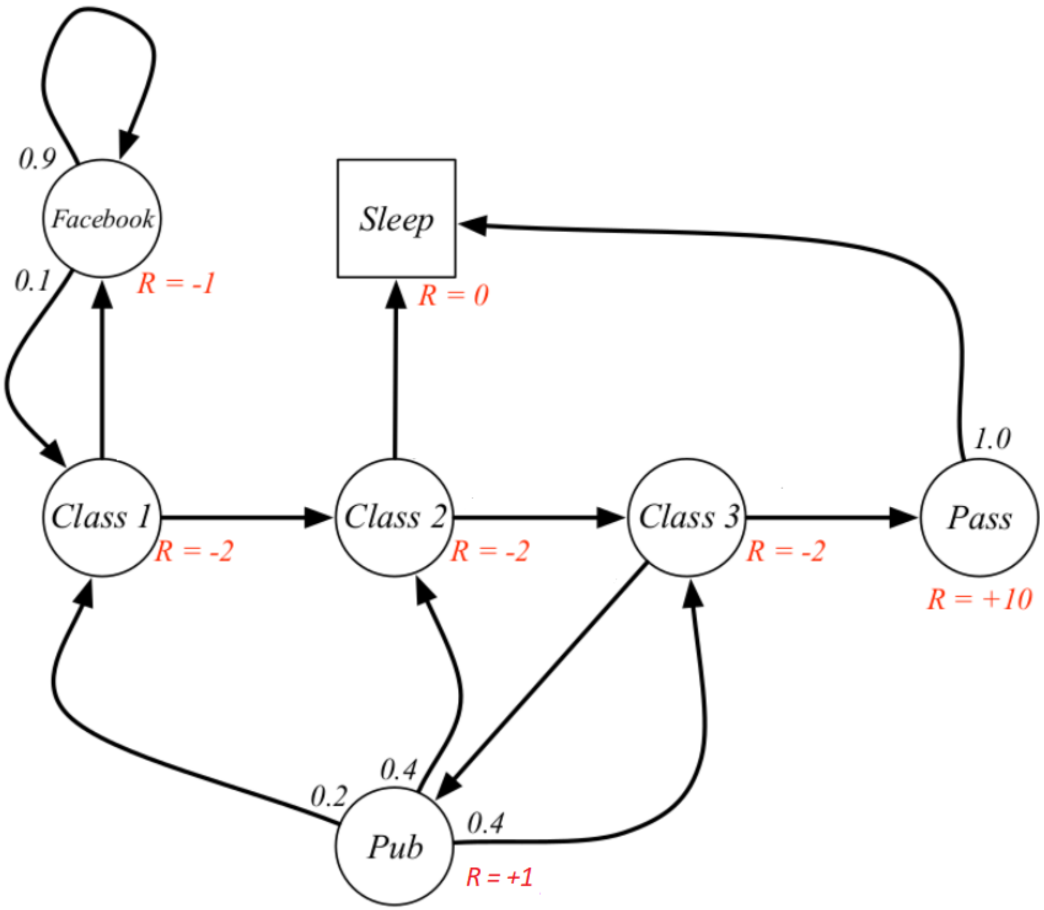
Agent-Environment interaction



Formal definition

- A Markov decision process is a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{R}, p)$ where:
    - $\mathcal{S}$ is the set of states
    - $\mathcal{A}$ is the set of actions
    - $\mathcal{R}$ is the set of rewards
    - $p(s', a|s, a) = P(S_t = s', R_t = r \mid S_{t-1} = s, A_{t-1} = a)$ governs the dynamics of the MDP.
- $S_t \in \mathcal{S}$, $R_t \in \mathcal{R}$ and $A_t \in \mathcal{A}$ for all t.
- $\sum_{s' \in \mathcal{S}, r \in \mathcal{R}} p(s', r|s, a) = 1$

Student MDP



0.9
Facebook
0.1
R = -1
Sleep
R = 0
1.0
Class 1
R = -2
Class 2
R = -2
Class 3
R = -2
Pass
R = +10
0.4
0.2
Pub
0.4
R = +1

Markovity

- State-transition probabilities:

$$p(s' \mid s, a) = P(S_t = s' \mid S_{t-1} = s, A_{t-1} = a) = \sum_{r \in \mathcal{R}} p(s', r \mid s, a)$$

- Expected rewards:

$$r(s, a) = \mathbb{E}\left[R_t \mid S_{t-1} = s, A_{t-1} = a\right] = \sum_{s' \in \mathcal{S}, r \in \mathcal{R}} p(s', r \mid s, a)$$

Objective/goal

- To maximize the expected sum of rewards $E(G_0)$:

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

- $\gamma \in [0, 1]$ is called the discount factor.
- For infinite episodes with bounded rewards, we must have $\gamma < 1$ so that $\sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$ converges.
- For finite episodes (γ can be 1, no discounting): $G_t = \sum_{k=0}^{T} \gamma^k R_{t+k+1}$.
- $G_t = R_{t+1} + \gamma G_{t+1}$.

Policy and Value functions

- Policy function $\pi(a|s)$: the probability that $A_t = a$ given that $S_t = s$
- State-value function for policy π:
  - this is the value of a state s under policy π, i.e. the expected return when starting in s and following π thereafter
  - $v_\pi(s) := \mathbb{E}_\pi [G_t \mid S_t = s]$
  - By convention, for terminal states, if any, their value is 0.
- Action-value function for policy π:
  - the value of taking action a in state s under policy π (and thereafter)
  - $q_\pi(s, a) := \mathbb{E}_\pi [G_t \mid S_t = s, A_t = a]$

Bellman equation (wrt a policy π)

$$
\begin{aligned}
v_\pi(s) &\doteq \mathbb{E}_\pi[G_t \mid S_t = s] \\
&= \mathbb{E}_\pi[R_{t+1} + \gamma G_{t+1} \mid S_t = s] \\
&= \sum_a \pi(a|s) \sum_{s'} \sum_r p(s', r | s, a) \Big[r + \gamma \mathbb{E}_\pi[G_{t+1} | S_{t+1} = s'] \Big] \\
&= \sum_a \pi(a|s) \sum_{s', r} p(s', r | s, a) \Big[r + \gamma v_\pi(s') \Big], \quad \text{for all } s \in \mathcal{S},
\end{aligned}
$$

- v_π is the solution to its own Bellman equation
- This is a system of $|\mathcal{S}|$ linear equations in $|\mathcal{S}|$ unknowns.

Policy evaluation (prediction)

- When $|S|$ is small, we can directly solve the linear equations.
- When the state-space is large, it is more efficient to do iteration using the following update formula:

$$\begin{aligned} v_{k+1}(s) &= \mathbb{E}_\pi \left[R_{t+1} + \gamma v_k \left(S_{t+1} \right) \mid S_t = s \right] \\ &= \sum_a \pi(a|s) \sum_{s',r} p(s', r|s, a)[r + \gamma v_k(s')] \end{aligned}$$

Iterative policy evaluation

Iterative Policy Evaluation, for estimating $V \approx v_\pi$

Input π, the policy to be evaluated
Algorithm parameter: a small threshold $\theta > 0$ determining accuracy of estimation
Initialize $V(s)$, for all $s \in \mathcal{S}^+$, arbitrarily except that $V(\textit{terminal}) = 0$

Loop:
  $\Delta \leftarrow 0$
  Loop for each $s \in \mathcal{S}$:
    $v \leftarrow V(s)$
    $V(s) \leftarrow \sum_a \pi(a|s) \sum_{s',r} p(s', r \,|\, s, a)\big[r + \gamma V(s')\big]$
    $\Delta \leftarrow \max(\Delta, |v - V(s)|)$
until $\Delta < \theta$

Optimal policies and value functions

- $\pi \geq \pi'$ if and only if $v_\pi(s) \geq v_{\pi'}(s)$ for all $s \in \mathcal{S}$.
- Optimal state-value function:

$$v_*(s) := \max_\pi v_\pi(s)$$

  for all s.
- Optimal action-value function:

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

  for all s, a.

Optimal Bellman equation (state-value)

$$
\begin{aligned}
v_*(s) &= \max_{a \in \mathcal{A}(s)} q_{\pi_*}(s, a) \\
&= \max_a \mathbb{E}_{\pi_*}[G_t \mid S_t = s, A_t = a] \\
&= \max_a \mathbb{E}_{\pi_*}[R_{t+1} + \gamma G_{t+1} \mid S_t = s, A_t = a] \\
&= \max_a \mathbb{E}[R_{t+1} + \gamma v_*(S_{t+1}) \mid S_t = s, A_t = a] \\
&= \max_a \sum_{s', r} p(s', r \mid s, a)\big[r + \gamma v_*(s')\big].
\end{aligned}
$$

Optimal Bellman equation (action-value)

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

- Both optimal Bellman equations are non-linear and cannot be solved explicitly.

Policy improvement

- Given the state-value function of a policy π, how do we find a new policy π' that is better than it?
- We can adopt the greedy approach, which does a one-step look-ahead (for each state s):

$$\begin{aligned}\pi'(s) &:= \arg\max_a q_\pi(s,a) \\ &= \arg\max_a \mathbb{E}\left[R_{t+1} + \gamma v_\pi(S_{t+1}) \mid S_t = s, A_t = a\right] \\ &= \arg\max_a \sum_{s',r} p(s',r \mid s,a)[r + \gamma\, v_\pi(s')]\end{aligned}$$

Policy iteration

- $\pi_0 \xrightarrow{E} v_{\pi_0} \xrightarrow{I} \pi_1 \xrightarrow{E} v_{\pi_1} \xrightarrow{I} \pi_2 \xrightarrow{E} \cdots \xrightarrow{I} \pi_* \xrightarrow{E} v_*$
- Here, $\xrightarrow{E}$ denotes policy evaluation and $\xrightarrow{I}$ denotes policy improvement.

Policy iteration

Policy Iteration (using iterative policy evaluation) for estimating $\pi \approx \pi_*$

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

2. Policy Evaluation
   Loop:
   $\quad \Delta \leftarrow 0$
   $\quad$ Loop for each $s \in \mathcal{S}$:
   $\quad\quad v \leftarrow V(s)$
   $\quad\quad V(s) \leftarrow \sum_{s',r} p(s', r | s, \pi(s))\big[r + \gamma V(s')\big]$
   $\quad\quad \Delta \leftarrow \max(\Delta, |v - V(s)|)$
   until $\Delta < \theta$ (a small positive number determining the accuracy of estimation)

3. Policy Improvement
   policy-stable $\leftarrow$ *true*
   For each $s \in \mathcal{S}$:
   $\quad$ *old-action* $\leftarrow \pi(s)$
   $\quad \pi(s) \leftarrow \operatorname{argmax}_a \sum_{s',r} p(s', r | s, a)\big[r + \gamma V(s')\big]$
   $\quad$ If *old-action* $\neq \pi(s)$, then *policy-stable* $\leftarrow$ *false*
   If *policy-stable*, then stop and return $V \approx v_*$ and $\pi \approx \pi_*$; else go to 2

Value iteration

- Combines policy evaluation and improvement into one step.
- Convert optimal Bellman equation into an update rule:

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

Value iteration

Value Iteration, for estimating $\pi \approx \pi_*$

Algorithm parameter: a small threshold $\theta > 0$ determining accuracy of estimation
Initialize $V(s)$, for all $s \in \mathcal{S}^+$, arbitrarily except that $V(\textit{terminal}) = 0$

Loop:
| $\Delta \leftarrow 0$
| Loop for each $s \in \mathcal{S}$:
| $\quad v \leftarrow V(s)$
| $\quad V(s) \leftarrow \max_a \sum_{s',r} p(s', r \,|\, s, a)\big[r + \gamma V(s')\big]$
| $\quad \Delta \leftarrow \max(\Delta, |v - V(s)|)$
until $\Delta < \theta$

Output a deterministic policy, $\pi \approx \pi_*$, such that
$\pi(s) = \arg\max_a \sum_{s',r} p(s', r \,|\, s, a)\big[r + \gamma V(s')\big]$

- Greedy policy with respect to optimal value function:

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