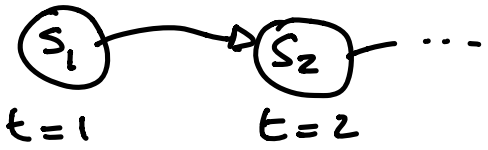
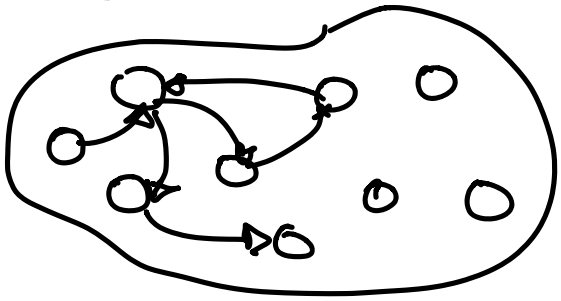
Markov Decision Processes (MDPs)

Markov processes (aka Markov chains)





Markov property (memoryless)

Def: A state S_t is Markov iff

$$\mathbb{P}(S_{t+1} \mid S_t) = \mathbb{P}(S_{t+1} \mid S_1, S_2 \cdots S_t)$$

"The future is independent of the past given the present"

State transition matrix

also transition probability matrix

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

$$\mathcal{P} = \begin{matrix} \\ s \downarrow \end{matrix} \overset{s' \rightarrow}{\begin{bmatrix} & \\ & \end{bmatrix}}$$

$\mathcal{P}_{ss'} \in [0, 1]$

sums to 1

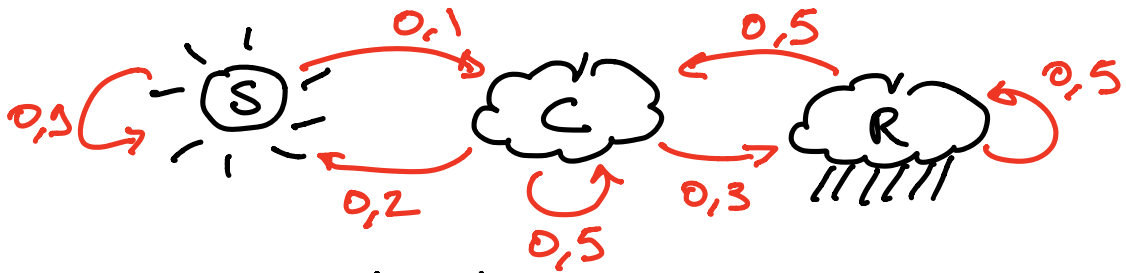
Definition of a Markov process

It is a sequence of random states $S_1, S_2 \ldots$ with Markov property.

A Markov process is fully defined by

- the finite set of states
- the state transition matrix

Example



$$P = \begin{array}{c} \\ S \\ C \\ R \end{array} \begin{array}{c} \begin{array}{ccc} S' & C' & R' \end{array} \\ \begin{bmatrix} 0,9 & 0,1 & 0 \\ 0,2 & 0,5 & 0,3 \\ 0 & 0,5 & 0,5 \end{bmatrix} \end{array}$$

each row sums to 1

$$S_t = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \begin{matrix} S \\ C \\ R \end{matrix} \quad : \quad \text{"sunny" state}$$

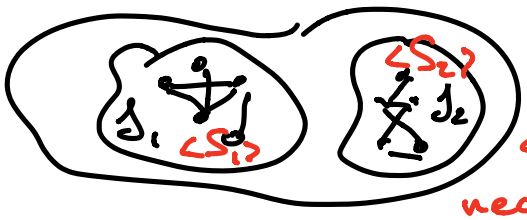
$$\sum_s P_{ss'} (S_t)_s \longrightarrow \begin{bmatrix} P(S_{t+1} = \text{sunny} \mid S_t) \\ \vdots \\ P(S_{t+1} = \text{rainy} \mid S_t) \end{bmatrix}$$

$$P(S_{t+k} \mid S_t) = (P^T)^k S_t$$

After a long time, we expect

$$\langle S \rangle = P^T \underbrace{\langle S \rangle}_{\text{mean distribution of states}}$$

This means that the mean distribution $\langle S \rangle$ is the eigenvector of P with eigenvalue 1.



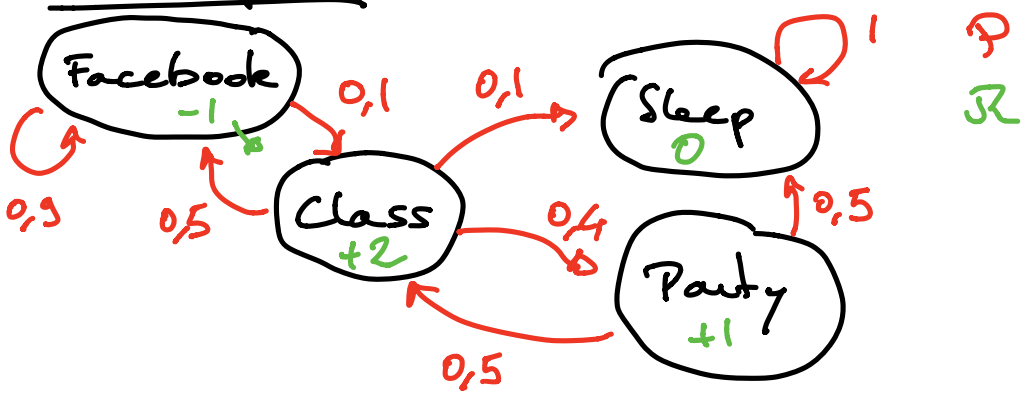
⟨S⟩ is not necessarily unique

$$\langle S \rangle = \begin{bmatrix} P(S = \text{sunny}) \\ \vdots \\ P(S = \text{rainy}) \end{bmatrix}$$

Markov reward processes

- Finite set of states S
- State transition matrix $\mathcal{P}_{ss'}$
- Reward function: $\mathcal{R}_s = \mathbb{E}[R_{t+1} \mid S_t = s]$
- Discount factor: $\gamma \in [0, 1]$

Example



Return

Def: the return G_t is the total discounted sum of future rewards:

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

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

Value function

Def: the **state** value function of a MRP is the expected return from state s

$$v(s) = \mathbb{E}(G_t \mid S_t = s)$$

Example

Facebook / Class / Party / Sleep

* $\gamma = 0$

$$G_t = R_{t+1}$$

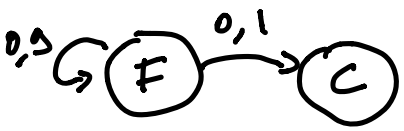
$$v(s) = \mathbb{E}(R_{t+1} \mid S_t = s)$$

$$= R_s$$

$v(F) = -1$; $v(C) = +2$

$v(P) = +1$; $v(S) = 0$

* $\gamma \neq 0$



$$v(F) = \mathbb{E}\left[G_t \mid S_t = F \right]$$

$$= \mathbb{E}\left[R_{t+1} + \gamma G_{t+1} \mid S_t = F \right]$$

$$= R_F + \gamma \mathbb{E}\left[G_{t+1} \mid S_{t+1} = F \right] \times$$

P_{FF} $\mathbb{P}\left[S_{t+1} = F \mid S_t = F \right]$

$v(C)$ $+\gamma \mathbb{E}\left[G_{t+1} \mid S_{t+1} = C \right]$ $v(F)$ P_{FC} $\mathbb{P}\left[S_{t+1} = C \mid S_t = F \right]$

$$v(F) = -1 + \gamma \left[v(F) \times 0,3 + v(C) \times 0,1 \right]$$

$$v(C) = +2 + \gamma \left[v(F) \times 0,5 + v(P) \times 0,4 + v(S) \times 0,1 \right]$$

$$v(P) = \text{———}$$

$$v(S) = 0 + \gamma\, v(S) \implies v(S) = 0$$

Bellman equation

$$v(s) = R_s + \gamma \sum_{s'} P_{ss'}\, v(s')$$

A Bellman equation link the value at a state s to the values of reachable states s' (from s).

Vector form : $V = \begin{bmatrix} v(S_1) \\ \vdots \\ v(S_n) \end{bmatrix}$

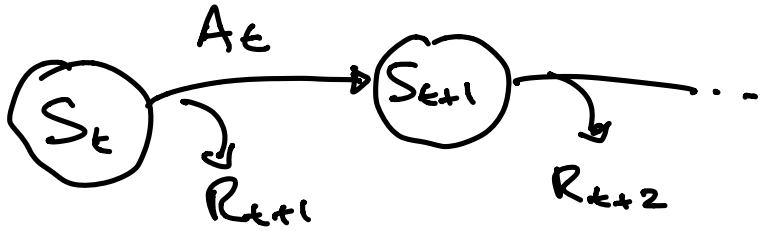
$$V = R + \gamma P V$$

which can be solved:

$$V = (I - \gamma P)^{-1} R$$

Markov Decision processes (MDPs)

- Finite set of states S
- Finite set of actions $\mathcal{A}$
- State transition matrix
  $$P^a_{ss'} \equiv \mathbb{P}(S_{t+1} = s' \mid S_t = s, A_t = a)$$
- Reward function
  $$R^a_s \equiv \mathbb{E}(R_{t+1} \mid S_t = s, A_t = a)$$
- Discount factor $\gamma \in [0, 1]$



Policy (strategy to choose the action)

Def

$$\pi(a \mid s) \equiv \mathbb{P}(A_t = a \mid S_t = s)$$

Rem: π can be deterministic: $a = \pi(s)$

π is time-independent and memoryless

Knowing π, the MDP becomes a MRP

$$\mathcal{P}^{\pi}_{ss'} = \sum_a \pi(a|s)\, \mathcal{P}^a_{ss'}$$

$$\mathcal{R}^{\pi}_{s} = \sum_a \pi(a|s)\, \mathcal{R}^a_s$$

Action-value function

state value : $v_\pi(s) = \mathbb{E}_\pi(G_t \mid S_t = s)$

action value : $q_\pi(s,a) = \mathbb{E}_\pi(G_t \mid S_t = s, A_t = a)$

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

Bellman equations

MRP : $v(s) = \mathcal{R}_s + \gamma \sum_{s'} \mathcal{P}_{ss'}\, v(s')$

MDP :

$$v_\pi(s) = \sum_a \pi(a|s) \left[\mathcal{R}^a_s + \gamma \sum_{s'} \mathcal{P}^a_{ss'}\, v_\pi(s') \right]$$

$$q_\pi(s,a) = \mathcal{R}^a_s + \gamma \sum_{s'} \mathcal{P}^a_{ss'}\, v_\pi(s')$$

$$q_\pi(s,a) = \mathcal{R}^a_s + \gamma \sum_{s'} \mathcal{P}^a_{ss'} \sum_{a'} \pi(a'|s')\, q_\pi(s',a')$$

Optimality

We want to find the "best" policy, which is the policy that maximizes $v(s)$ or $q(s,a)$.

Optimal values:

$$\begin{cases} v_*(s) = \max_\pi v_\pi(s) \\ q_*(s,a) = \max_\pi q_\pi(s,a) \end{cases}$$

The goal of reinforcement learning is to find the policy π^* which maximizes $v_\pi(s)$.

theorem: For any MDP (fully observable), there is an optimal policy π^* that is deterministic.

proof (idea):

π' optimal but not deterministic.

$$q_{\pi'}(s, a_1) = q_{\pi'}(s, a_2)$$

we can always choose arbitrarily a_1

$$\pi^*(a|s) = \begin{cases} 1 & \text{if } a = \text{argmax } q_{\pi'}(s,a) \\ 0 & \text{otherwise} \end{cases}$$

this deterministic policy π^* has the same value as π':

$$v_{\pi^*}(s) = \max_a q_{\pi'}(s,a) = v_{\pi'}(s)$$

Bellman optimality equations

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

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

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

$$v_*(s) = \max_a \left[\mathcal{R}_s^a + \gamma \sum_{s'} \mathcal{P}_{ss'}^a v_*(s') \right]$$

$$\pi^*(s) = \operatorname{argmax}_a q_*(s,a)$$

Dynamic programming

iterative process

optimize / calculate v_* for given π

2 axes:

- Prediction: given π, we want $v_\pi(s)$
- Control: we want π^* and $v_*(s)$

Value iteration (prediction).

Recall that

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

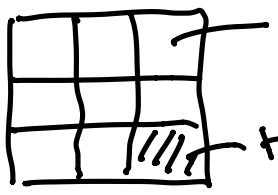
(Bellman eq. for v_π)

$k=0$ $k=1$

$v_0(s) \longrightarrow v_1(s) \longrightarrow \cdots \longrightarrow v_\pi(s)$

$$v_{k+1}(s) = \sum_a \pi(a|s) \left[\mathcal{R}_s^a + \gamma \sum_{s'} \mathcal{P}_{ss'}^a v_k(s') \right]$$

Example



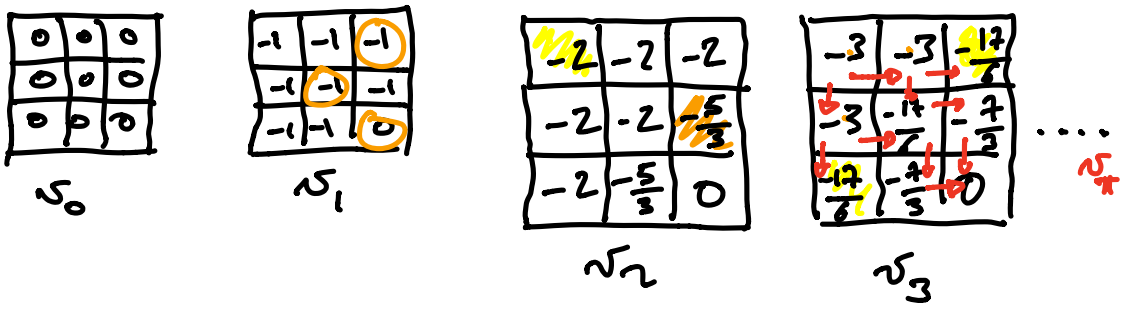
reward exiting: -1

actions $\uparrow \rightarrow \downarrow \leftarrow$ when possible.

terminal state: goal ($r=0$)

π: random action from allowed moves

$\gamma = 1$



Policy iteration

- Initialize policy π (random policy, for ex.)
- Use value iteration to find $v_\pi(s)$ (loop V.I.)
- Update π by using a greedy policy

$$a = \pi_{greedy}(s) = \underset{\text{reachable } s'}{\operatorname{argmax}}\ v_\pi(s')$$

loop policy iteration

This process converges towards π^*