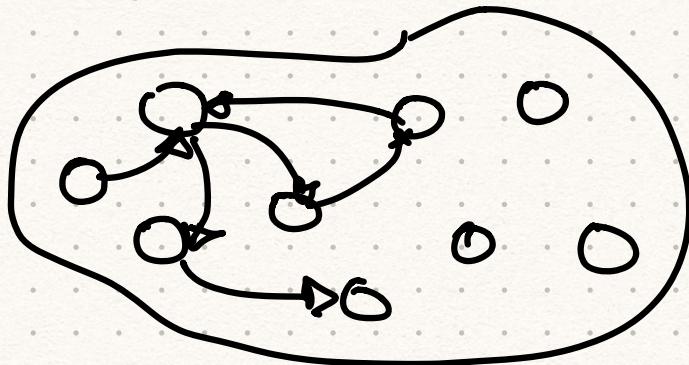
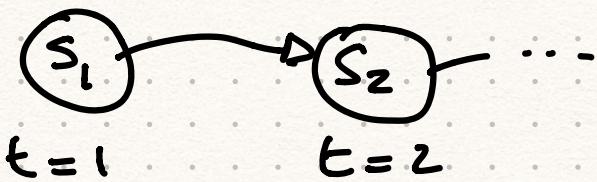


Markov Decision Processes (MDPs)

Markov processes (aka Markov chains)



Markov property (memoryless)

Def: A state \$S_t\$ is Markov if

$$P(S_{t+1} | \underline{S_t}) = P(S_{t+1} | \underline{S_1, S_2, \dots, S_t})$$

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

State transition matrix

also transition probability matrix

$$P_{ss'} = P(S_{t+1} = s' | S_t = s)$$

$$P = \begin{matrix} & \xrightarrow{s'} \\ \downarrow s & \left[\begin{array}{c} \text{---} \\ \text{---} \end{array} \right] \end{matrix} \quad P_{ss'} \in [0, 1]$$

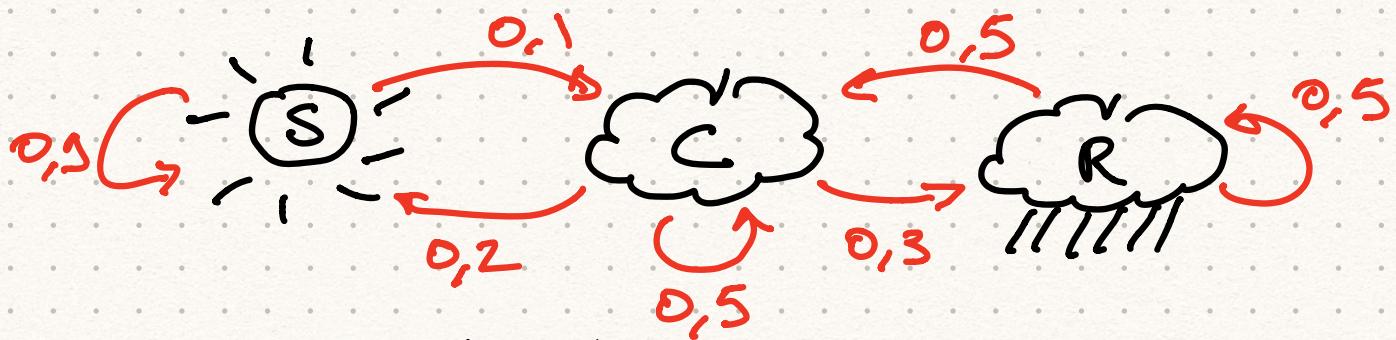
sums to 1

Definition of a Markov process

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

- A Markov process is fully defined by
- \mathcal{S} the finite set of states
 - P the state transition matrix

Example



$$P = \begin{matrix} & S' & C' & R' \\ S & \left[\begin{matrix} 0,1 & 0,1 & 0 \end{matrix} \right] \\ C & \left[\begin{matrix} 0,2 & 0,5 & 0,3 \end{matrix} \right] \\ R & \left[\begin{matrix} 0 & 0,5 & 0,5 \end{matrix} \right] \end{matrix}$$

each row
sums to 1

$$S_t = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}_S^T : \text{"sunny" state}$$

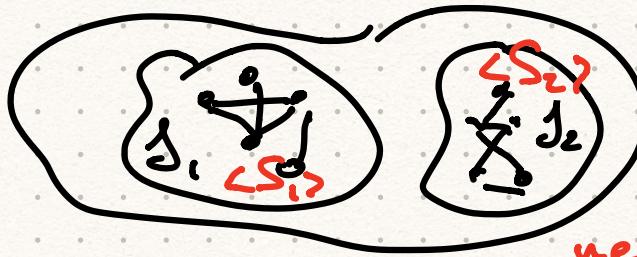
$$\sum_s \beta_{ss'} (S_t)_s \rightarrow \begin{bmatrix} P(S_{t+1} = \text{sunny} | S_t) \\ \vdots \\ P(S_{t+1} = \text{rainy} | S_t) \end{bmatrix}$$

$$P(S_{t+k} | S_t) = (\beta^\top)^k S_t$$

After a long time, we expect

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

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



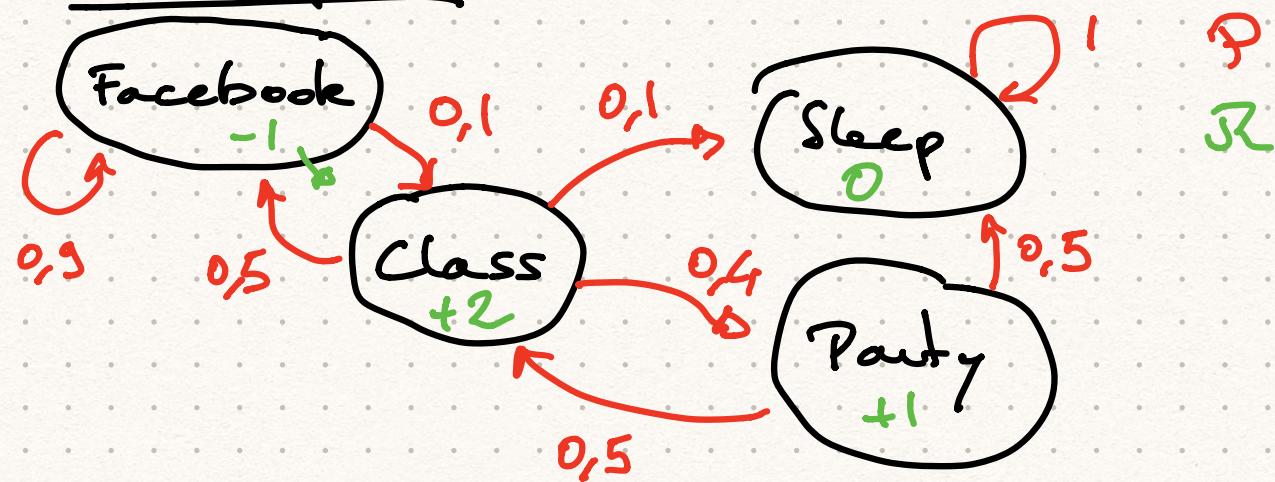
\int
 $\langle S \rangle$ is not necessarily unique

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

Markov reward processes

- Finite set of states S
- State transition matrix $P_{SS'}$
- Reward function: $R_S = \mathbb{E}[R_{t+1} | S_t = s]$
- Discount factor: $\gamma \in [0, 1]$

Example



Return

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

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

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



Value function

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

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

Example

Facebook / Class / Party / Sleep

* $\gamma = 0$

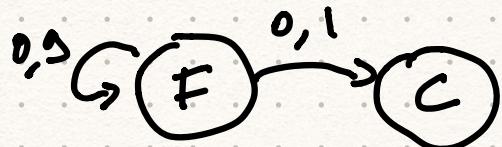
$$G_t = R_{t+1}$$

$$\begin{aligned} v(s) &= \mathbb{E}(R_{t+1} | S_t = s) \\ &= R_s \end{aligned}$$

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

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

* $\gamma \neq 0$



$$\begin{aligned}
 v(F) &= \mathbb{E} [G_t \mid S_t = F] \\
 &= \mathbb{E} [R_{t+1} + \gamma G_{t+1} \mid S_t = F] \\
 &= R_F + \gamma \mathbb{E} [G_{t+1} \mid S_{t+1} = F] \times \\
 &\quad \mathbb{P}[S_{t+1} = F \mid S_t = F] \\
 &\quad + \gamma \mathbb{E} [G_{t+1} \mid S_{t+1} = C] \mathbb{P}[S_{t+1} = C \mid S_t = F]
 \end{aligned}$$

$$v(F) = -1 + \gamma [v(F) \times 0,9 + v(C) \times 0,1]$$

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

$$v(P) = \underline{\hspace{2cm}}$$

$$v(S) = 0 + \gamma v(S) \Rightarrow v(S) = 0$$

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 : $\mathbf{V} = \begin{bmatrix} v(s_1) \\ \vdots \\ v(s_n) \end{bmatrix}$

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

which can be solved :

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

Markov Decision processes (MDPs)

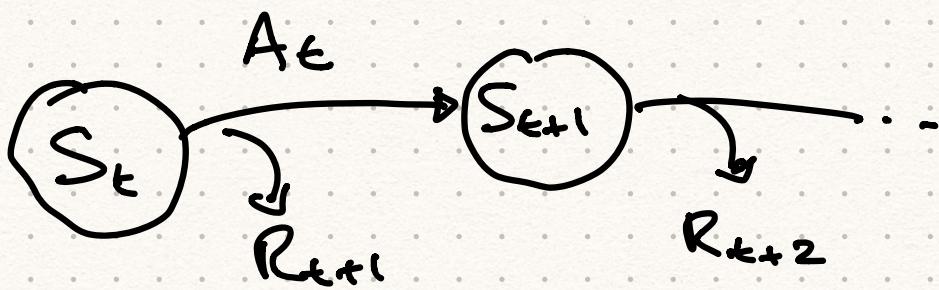
- Finite set of states \mathcal{S}
- Finite set of actions \mathcal{A}
- State transition matrix

$$P_{ss'} \equiv P(S_{t+1} = s' | S_t = s, A_t = a)$$

- Reward function

$$R_s^a \equiv E(R_{t+1} | S_t = s, A_t = a)$$

- Discount factor $\gamma \in [0, 1]$



Policy (strategy to choose the action)

Def

$$\pi(a|s) \equiv P(A_t = a | S_t = s)$$

Rem : π can be deterministic : $a = \pi(s)$
 π is time-independent and memoryless

Knowing π , the TD π becomes a TDP

$$\left| \begin{array}{l} \mathcal{P}_{ss'}^\pi = \sum_a \pi(a|s) \mathcal{P}_{ss'}^a \\ \mathcal{R}_s^\pi = \sum_a \pi(a|s) \mathcal{R}_s^a \end{array} \right.$$

Action-value function

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

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

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

Bellman equations

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

TDP,

$$v_\pi(s) = \sum_a \pi(a|s) \left[R_s^a + \gamma \sum_{s'} P_{ss'}^a v_\pi(s') \right]$$

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

$$q_\pi(s, a) = R_s^a + \gamma \sum_{s'} P_{ss'}^a \sum_{a'} \pi(a'|s') q_\pi(s', a')$$

Optimality

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

Optimal values:

$$v_*(s) = \max_{\pi} v_{\pi}(s)$$

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

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

Theorem: \Rightarrow any TD-P (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 ,

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

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

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

$$q_*(s, a) = R_s^a + \gamma \sum_{s'} P_{ss'}^a v_*(s')$$

$$q_*(s, a) = R_s^a + \gamma \sum_{s'} P_{ss'}^a \max_{a'} q_*(s', a')$$

$$v_*(s) = \max_a \left[R_s^a + \gamma \sum_{s'} P_{ss'}^a v_*(s') \right]$$

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

Dynamic programming

iterative process

optimize
calculate v_{π} for given π

2 uses:

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

Value iteration (prediction).

Recall that

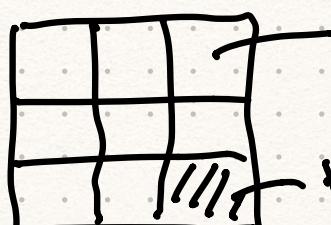
$$v_{\pi}(s) = \sum_a \pi(a|s) \left[R_s^a + \gamma \sum_{s'} P_{ss'}^a v_{\pi}(s') \right]$$

(Bellman eq. for v_{π})

$$v_0(s) \xrightarrow{k=0} v_1(s) \xrightarrow{k=1} \dots \xrightarrow{} v_{\pi}(s)$$

$$v_{k+1}(s) = \sum_a \pi(a|s) \left[R_s^a + \gamma \sum_{s'} P_{ss'}^a v_k(s') \right]$$

Example



reward exiting: -1

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

terminal state: goal ($r=0$)

π : random action from allowed moves
 $\gamma = 1$

0	0	0
0	0	0
0	0	0

v_0

-1	-1	-1
-1	-1	0
-1	0	0

v_1

-2	-2	-2
-2	-2	$\frac{5}{3}$
-2	$\frac{5}{3}$	0

v_2

-3	-3	$\frac{-17}{6}$
-3	$\frac{-17}{6}$	$\frac{7}{2}$
$\frac{-17}{6}$	$\frac{7}{2}$	0

v_3

...
 v_T

Policy iteration

- Initialize policy π (random policy, for ex.)
- Use value iteration to find $v_{\pi}(s)$ loop
VI.
- Update π by using a greedy policy
 $a = \pi_{\text{greedy}}(s) = \underset{\text{reachable } s'}{\arg \max} v_{\pi}(s')$

loop
policy iteration

This process converges towards π^*