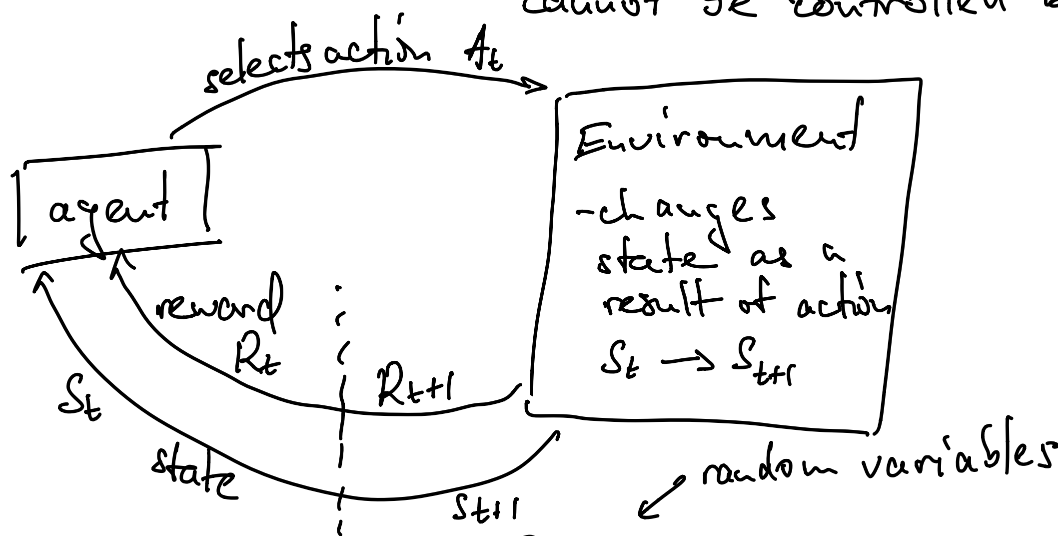
Markov Decision Processes (MDPs)

- Agent - Environment Interface

<u>agent</u>: learner / decision maker

<u>environment</u> (env.): everything outside agent; cannot be controlled by agent



<u>defs.</u>: 1) state space: $\mathcal{S} \ni S_t$

2) action space: $\mathcal{A} \ni A_t$ / allowed actions may depend on state

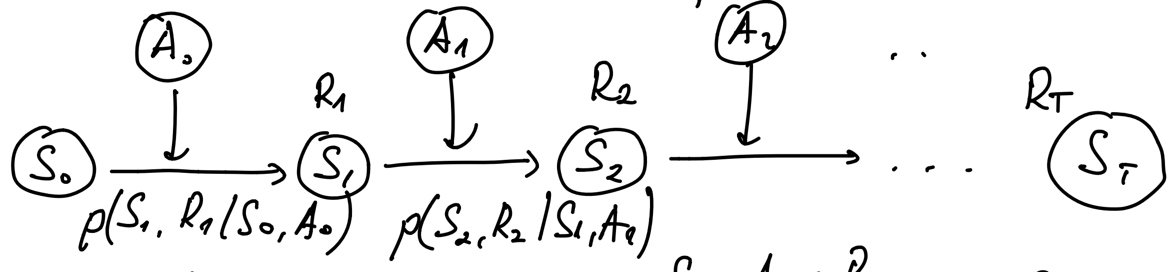
3) reward space: $\mathcal{R} \ni R_t$

- <u>trajectory</u>: $\underbrace{S_0 A_0}_{t=0} R_1 \underbrace{S_1 A_1}_{t=1} R_2 S_2 \dots$

step: $t=0$ $t=1$ ...

→ agent finds env. in the state S_0,
chooses action A_0,
which causes the env. to transition to state S_1,
and the agent receives a reward R_1 as a consequence

– finite Markov decision process (MDP)



- a Markov process on $\mathcal{S} \times \mathcal{A} \times \mathcal{R}$ space
- provide a mathematical abstraction for the problem: goal-directed learning from interactions
- the transition prob. p depends only on the preceding state and action, not on the history

$$p(s', r | s, a) := \Pr(S_t = s', R_t = r \mid S_{t-1} = s, A_{t-1} = a)$$

for a $s, s' \in \mathcal{S}$, $a \in \mathcal{A}(s)$, $r \in \mathcal{R}$

$\rightarrow$ defines the dynamics of the MDP

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

– state transition probability

$$p(s' | s, a) := \Pr(S_t = s' \mid S_{t-1} = s, A_{t-1} = a)$$

$$= \sum_{r \in \mathcal{R}} p(s', r | s, a)$$

(marginalize / integrate out rewards)

→ if rewards are stochastic:

- $r(s,a) := \mathbb{E}_p[R_t \mid S_{t-1} = s, A_{t-1} = a]$

$$= \sum_{r\in R}\sum_{s'\in S} r\, p(s',r \mid s,a)$$

<u>expected reward</u> for state-action pair (s,a)

- $r(s,a,s') := \mathbb{E}_p[R_t \mid S_{t-1} = s, A_{t-1} = a, S_t = s']$

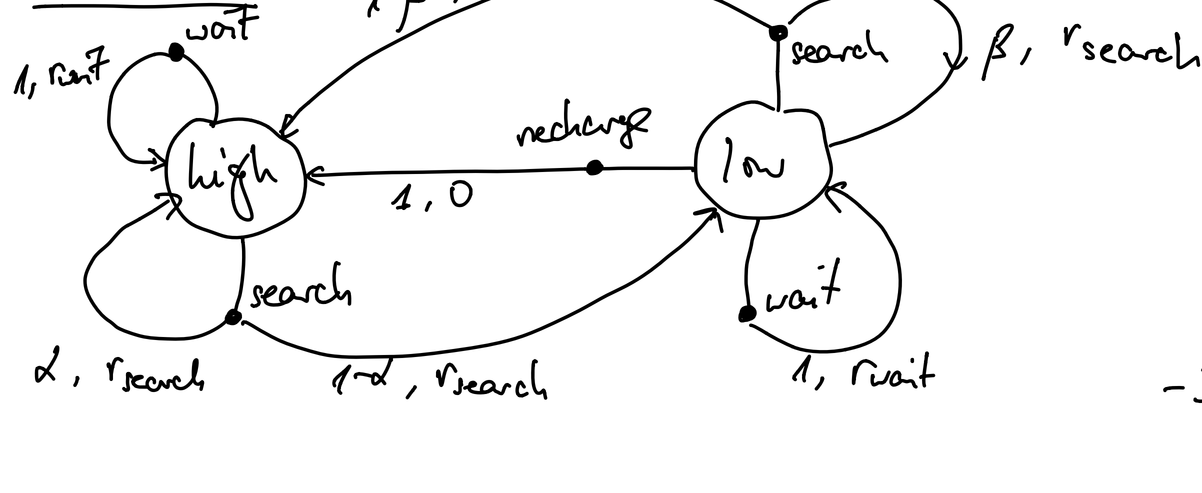
$$= \frac{\sum_{r\in R} r\, p(s',r \mid s,a)}{\sum_{r\in R} p(s',r \mid s,a)} = \sum_{r\in R} r\, \frac{p(s',r \mid s,a)}{p(s' \mid s,a)}$$

— <u>Example</u>: (recycling robot)

<u>goal</u>: teach a robot to collect empty soda cans

- state space: $S = \{\text{low}, \text{high}\}$ models battery level
- action space: $A(\text{high}) = \{\text{search}, \text{wait}\}$
  $A(\text{low}) = \{\text{search}, \text{wait}, \text{recharge}\}$
- reward space: $R = \{r_{\text{search}}, r_{\text{wait}}, 0, -3\}$

<u>Environment</u>:



s	a	s'	$p(s'\|s,a)$	$r(s,a,s')$
high	search	high	α	r_{search}
high	search	low	$1-\alpha$	r_{search}
high	wait	high	1	r_{wait}
high	wait	low	0	–
low	search	high	$1-\beta$	-3
low	search	low	β	r_{search}
low	wait	low	1	r_{wait}
low	wait	high	0	–
low	recharge	low	0	–
low	recharge	high	1	0

- rewards: $R_t \in \mathcal{R} \subset \mathbb{R}$
  - communicate to agent <u>what</u> it has to achieve, not how to achieve it
  - instantaneous
  - goal for agent: maximize <u>total</u> <u>expected</u> <u>return</u>

→ sequence of rewards within trajectory:

$R_1, R_2, R_3, \dots, R_T$

<u>return</u>: $G_t = R_{t+1} + R_{t+2} + R_{t+3} + \dots + R_T$

reward to-go

– two kind of tasks:

→ episodic / finite-horizon tasks:

- agent learns in multiple repeated trials called episodes (e.g. plays of a game)

→ each episode is broken down in steps labelled by t

$$\underbrace{S_0 \xrightarrow{A_0}}_{\text{step } 0} \underbrace{S_1 \xrightarrow{A_2}}_{\text{step } 1} S_2 \to \dots \underbrace{S_t \xrightarrow{A_t}}_{\text{step } t} \dots \underbrace{S_{T-1} \xrightarrow{A_{T-1}} S_T}_{\text{step } T-1}$$

episode terminates

→ subsequent episodes begin independently of how current episode ended

→ return: $G_t := \sum_{k=t+1}^{T} R_k$

→ non-episodic / infinite-horizon tasks

– no episodes, continue forever

problem: $G_t = \infty$

→ way out: introduce discounted rewards

- agent cares more about near-term rewards!

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

$$= \sum_{k=0}^{\infty} \gamma^k R_{t+1+k} \quad , \gamma \in [0,1] \text{ discount factor}$$

$\Rightarrow G_t < \infty$ if $\{R_k\}_k$ is bounded

(e.g. $R_k = 1 \ \forall k \Rightarrow G_t = \sum_{k=0}^{\infty} \gamma^k = \frac{1}{1-\gamma}$)
geometric series

— unified notation:

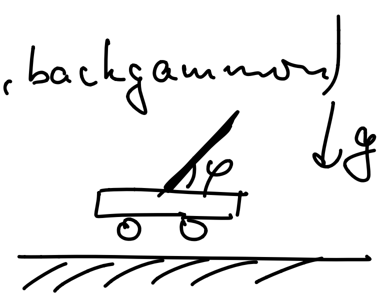
$$G_t := \sum_{k=0}^{T} \gamma^{k-t-1} R_k \ : \ \begin{cases} T = \infty & : \text{non-episodic} \\ T < \infty & : \text{episodic} \\ \gamma = 1 & \end{cases}$$

$$S_1 \xrightarrow{R_1} S_2 \xrightarrow{R_2} S_2 \xrightarrow{R_3} S_3 \longrightarrow \blacksquare \quad R_4 = 0, \ R_5 = 0$$

Examples:

1) finite-horizon tasks
   - → recycling robot
   - → solve a maze
   - → board games (chess, backgammon)

2) infinite-horizon tasks:
   - → cart pole problem: (pole-balancing)



   - → teach a robot to walk (forever)

Policies & Value Functions

How does the RL agent choose actions?

idea: agent observes state S_t of the env.
Can we correlate states & expected returns?

value function: estimate "how good" it is for the env. to be in a given state

note: future rewards depend on what actions the agent will take

→ value functions are defined wrt particular ways of acting, called policies

policy: $\pi : \mathcal{A} \times \mathcal{S} \longrightarrow [0,1]$

$$(a, s) \longmapsto \pi(a|s)$$

probability to take action a in the state s

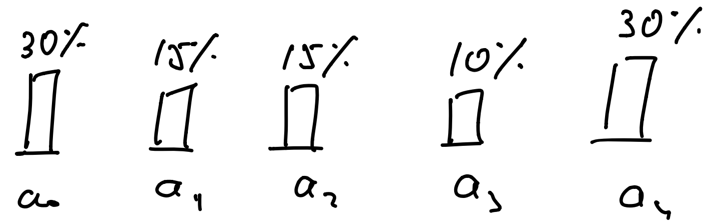
$$\sum_{a \in \mathcal{A}(s)} \pi(a|s) = 1 \quad \text{(normalized)}$$

1) deterministic policy: ("delta"-function policy)

$$\pi(a|s) = \delta_{a,a_0} = \begin{cases} 1, & \text{if } a = a_0 \\ 0, & \text{otherwise} \end{cases}$$

2) stochastic policy has weight on multiple actions.

e.g.



– expectation value of R_{t+1} under π

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

– value function (of a policy π)

$$v_\pi : S \longrightarrow \mathbb{R}$$
$$s \longmapsto v_\pi(s)$$

$$v_\pi(s) := \mathbb{E}_\pi[G_t \mid S_t = s] = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid S_t = s\right]$$

– expected return starting from a state s under the policy π

$\mathbb{E}_\pi[\cdot]$: expectation of a random variable given that agent follows policy π at every step t in the episode!

– recursive relation for $v_\pi(s)$

<u>recall</u>: $G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 R_{t+4} + \dots$

$$= R_{t+1} + \gamma \underbrace{\left(R_{t+2} + \gamma R_{t+3} + \gamma^2 R_{t+4} + \dots \right)}_{=: G_{t+1}}$$

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

$$= \mathbb{E}_\pi [R_{t+1} + \gamma G_{t+1} \mid S_t = s]$$

$$= \mathbb{E}_\pi [R_{t+1} \mid S_t = s] + \gamma \mathbb{E}_\pi [G_{t+1} \mid S_t = s]$$

$$= \sum_{a \in A(s)} \pi(a|s) \sum_{r \in R} \sum_{s' \in S} p(s', r | s, a)\, r$$

$$+ \gamma \sum_{a \in A(s)} \pi(a|s) \sum_{r \in R} \sum_{s' \in S} p(s', r | s, a) \underbrace{\mathbb{E}_\pi [G_{t+1} \mid S_{t+1} = s']}_{= v_\pi(s')}$$

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

relates $v_\pi(s)$ & $v_\pi(s')$

→ <u>Bellman equation</u> for v_π:

$$\boxed{v_\pi(s) = \sum_a \pi(a|s) \sum_{r \in R} \sum_{s' \in S} p(s', r | s, a) \left[r + \gamma v_\pi(s') \right]}$$

- averages all possibilities for trajectories weighting each by its probability for occuring

- the value of the starting state $v(s)$ is equal to the discounted value of the expected next state, plus the reward expected along the trajectory
- v_π is the <u>unique</u> solution of the Bellman eq. (finite MDPs)

- <u>Action-value functions</u>: q-functions (of a policy π)

$$q_\pi : \mathcal{S} \times \mathcal{A} \longrightarrow \mathbb{R}$$
$$(s, a) \longmapsto q_\pi(s, a)$$

$$q_\pi(s,a) := \mathbb{E}_\pi \left[G_t \mid S_t = s, A_t = a \right]$$

→ the expected return starting in a state s and taking the action a, and following the policy π afterwards

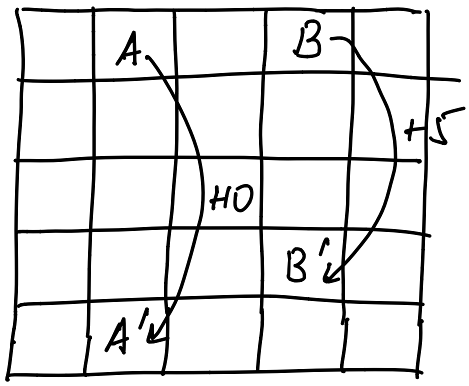
show (HW):

$$q_\pi(s,a) = \sum_{r \in \mathcal{R}} \sum_{s' \in \mathcal{S}} p(s', r \mid s, a) \left[r + \gamma v_\pi(s') \right]$$

- relation between $v_\pi(s)$ & $q_\pi(s,a)$?

$$v_\pi(s) = \sum_{a \in \mathcal{A}(s)} \pi(a \mid s) \, q_\pi(s,a)$$

Example: Gridworld: $r(s,a,s')$



rewards: $r(A, :, A') = +10 \quad \forall a \in A$

$r(B, :, B') = +5 \quad \forall a \in A$

$r(s_{boundary}, a_{outside}, s') = -1$

$r(s, a, s') = 0$, otherwise

states: squares on grid

actions: $\leftarrow \updownarrow \rightarrow$; cannot move across boundaries

policy: $\pi(a|s) = \frac{1}{|A|} = \frac{1}{4}$ equiprobable random policy

$\rightarrow$ value function $v_\pi(s)$ for $\gamma = 0.9$

3.3	8.8	4.4	5.3	1.5
1.5	3.0	2.3	1.9	0.5
0.1	0.7	0.7	0.4	-0.4
-1.0	-0.4	-0.4	-0.6	-1.2
-1.9	-1.3	-1.2	-1.4	-2.0

Optimal policies & value functions

- ordering relation for policies:

$\pi \geq \pi'$, iff $v_\pi(s) \geq v_{\pi'}(s) \quad \forall s \in S$

there exists (at least one) policy that is better than or equal to all other policies

$\Rightarrow$ optimal policy: π_*

- optimal value function: v_*

$\rightarrow$ value function associated with π_*

$$v_*(s) := \max_\pi v_\pi(s) \quad \forall s \in S$$

- optimal q-function: q_*

$$q_*(s,a) := \max_\pi q_\pi(s,a) \quad \forall s \in S, \ \forall a \in A$$

$$\Rightarrow q_*(s,a) = \mathbb{E}\left[R_{t+1} + v_*(S_{t+1}) \mid S_t = s, A_t = a \right]$$

- relation: $\boxed{v_*(s) = \max_{a \in A(s)} q_*(s,a)}$

$\rightarrow$ note: optimal policy π_* may not be unique but v_* & q_* are unique

- Bellman optimality equation for $v_*(s)$:

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

$$= \max_{a \in A(s)} \mathbb{E}_{\pi_*} [G_t \mid S_t = s, A_t = a]$$

$$= \max_{a \in A(s)} \mathbb{E}_{\pi_*} [R_{t+1} + \gamma G_{t+1} \mid S_t = s, A_t = a]$$

$$= \max_{a \in A(s)} \mathbb{E} [R_{t+1} + \gamma \underbrace{v_{\pi_*}(S_{t+1})}_{= v_*(S_{t+1})} \mid S_t = s, A_t = a]$$

$$= \max_{a \in A(s)} \sum_{s',r} p(s', r \mid s, a) [r + \gamma v_*(s')]$$

→ if v_* is known, then an optimal policy can be defined by taking all possible actions from s, and comparing the $v_*(s')$ values

=> greedy policy (deterministic)

→ v_* turns the total expected return to a local quantity, available at each step.

=> a one-step-ahead seach with v_* yields long-term optimal actions

– Bellman optimality equation for $q_*(s,a)$

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

→ resolution over the actions a

$$\pi_*(a|s) = \underset{a \in A(s)}{\operatorname{argmax}}\ q_*(s,a) \quad \text{optimal greedy policy}$$

→ no need for a one-step-ahead search

– Remarks:
→ exact solution to Bellman optimality eq.'s nearly feasible, since
1) do not know transition prob. $p(s', r \mid s, a)$ (→ use MC sampling to find high-probability trajectories)
2) state space S is often exponentially large → cannot find $v_*(s)$, $q_*(s,a)$ $\forall s, a$

- **Example**: optimal policy for Gridworld: ($\gamma = 0.9$)

22.8	22.4	22.0	19.4	17.5
19.8	22.0	19.8	17.8	16.0
17.8	19.8	17.8	16.0	14.4
16.0	17.8	16.0	14.4	13.0
14.4	16.0	14.4	13.0	11.7

$v_*(s)$

→	←↑→	←	←↑→↓	←
↑→	↑	←↑	←	←
↑→	↑	←↑	←↑	←↑
↑→	↑	←↑	←↑	←↑
↑→	↑	←↑	←↑	←↑

π_*