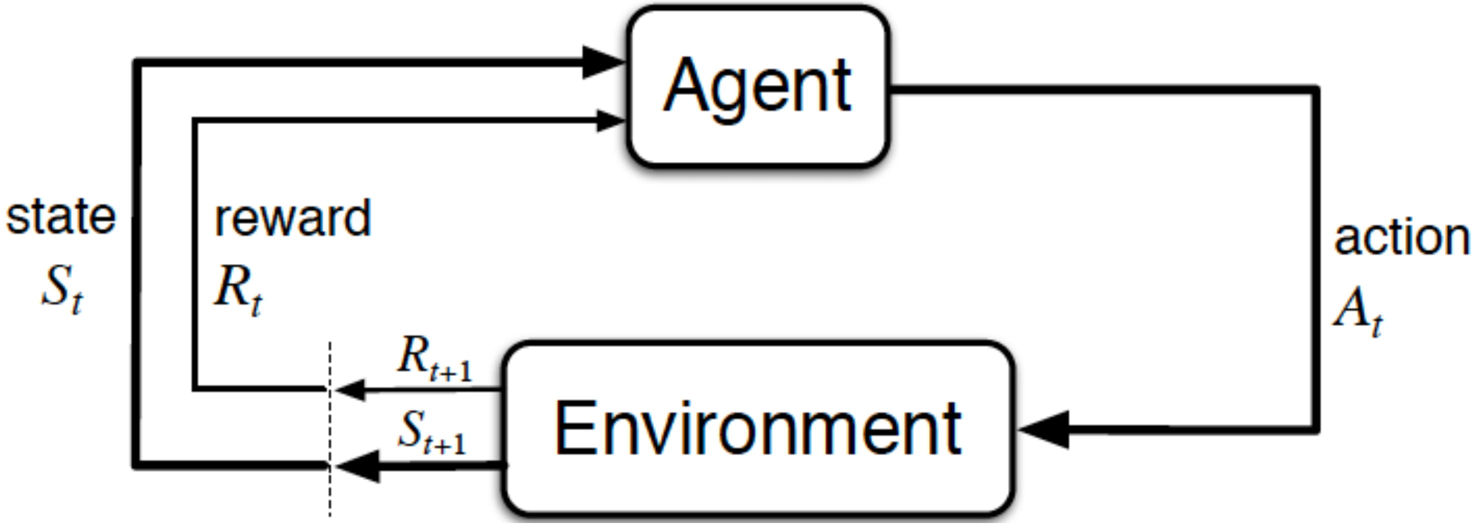
Finite Markov Decision Processes

- Formalisation of sequential decision making
- Trade-off between immediate and delayed reward.
- MDP frame the problem of learning from interaction to achieve a goal
- Actions are the choices made by the agent based on the states of the environment and the rewards are the basis for evaluating the choices.

Agent–Environment Interface

- Trajectory of states, actions and rewards: $S_0, A_0, R_1, S_1, A_1, R_2, S_2, A_2, R_3, \ldots$
- MDP dynamics: $p(s', r | s, a) \doteq \Pr\{S_t = s', R_t = r \mid S_{t-1} = s, A_{t-1} = a\},$



Goals and Rewards

- The goal of the agent is formalised in terms of a reward signal (should not impart prior knowledge)
- Reward hypothesis: goals and purposes can be thought of as the maximisation of the expected value of the cumulative sum of the reward.

Returns and Episodes

- Episodic tasks have a natural terminal state
- Continuing tasks
- Expected discounted return:

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

where γ is a parameter, $0 \leq \gamma \leq 1$, called the *discount rate*.

Policies and Value Functions

- Policy: mapping from states to probabilities of selecting each possible action : $\pi(a|s)$
- State value function: expected return when starting in state s and following policy π thereafter

$$v_\pi(s) \doteq \mathbb{E}_\pi[G_t \mid S_t = s] = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \,\middle|\, S_t = s\right], \text{ for all } s \in \mathcal{S},$$

- Action-value function: expected return when starting in state s, taking action a and then following policy π thereafter

$$q_\pi(s,a) \doteq \mathbb{E}_\pi[G_t \mid S_t = s, A_t = a] = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \,\middle|\, S_t = s, A_t = a\right].$$

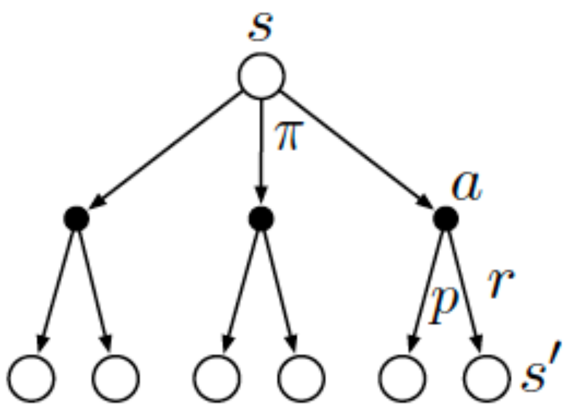
Bellman equations

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

- Value functions have recursive relationships

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

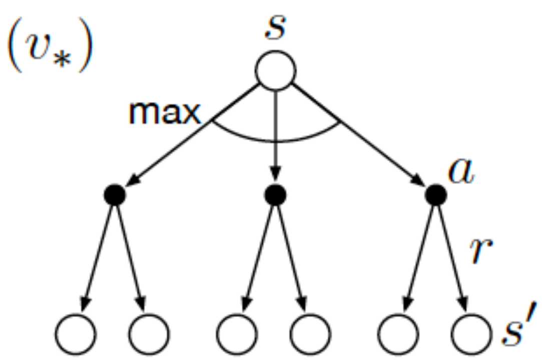
- The value of a state equals the discounted value of the expected next state, plus the reward expected along the way.



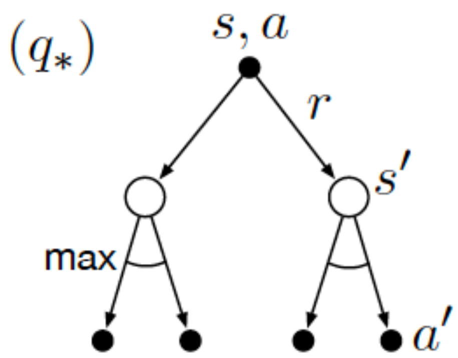
Backup diagram for v_π

Optimal Value Functions

- For optimal policies: $v_*(s) \doteq \max_\pi v_\pi(s), \quad q_*(s,a) \doteq \max_\pi q_\pi(s,a),$
- Bellman optimality equations



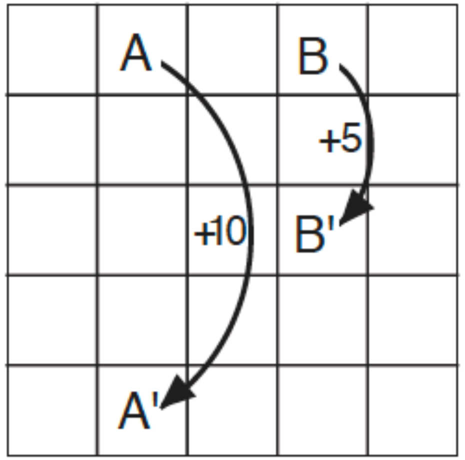
$$\begin{aligned} v_*(s) &= \max_{a \in \mathcal{A}(s)} q_{\pi_*}(s,a) \\ &= \max_a \mathbb{E}_{\pi_*}[G_t \mid S_t = s, A_t = a] \\ &= \max_a \mathbb{E}[R_{t+1} + \gamma v_*(S_{t+1}) \mid S_t = s, A_t = a] \\ &= \max_a \sum_{s',r} p(s',r|s,a)\big[r + \gamma v_*(s')\big]. \end{aligned}$$



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

Optimal Policies

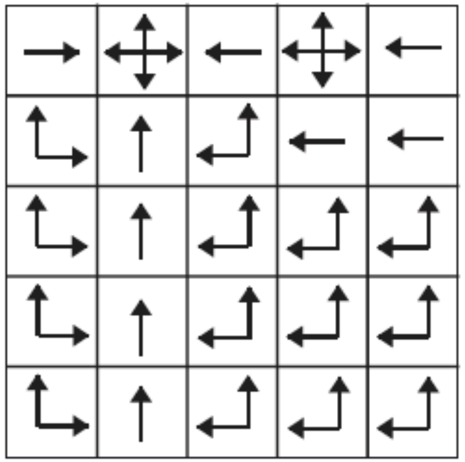
- Acting greedy with respect to v_* yields an optimal policy (one-step-ahead search yields the long-term optimal actions)



Gridworld

22.0	24.4	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_*



π_*

- The optimal action is the action that maximises $q_*(s,a)$

Optimality and Approximation

- Solving the Bellman optimality equations is an exhaustive search, looking ahead at all possibilities and computing the probabilities and expected rewards, at extreme computational cost.
- Reinforcement learning methods approximately solving the Bellman optimality equation using actual experienced transitions as we are unlikely to know the dynamics of the environment.
- Online nature of reinforcement learning makes it possible to approximate optimal policies by putting in more effort into learning to make good decisions for frequently encountered states