3.1: The Agent-Environment Interface

In a finite MDP: sets of states, actions, rewards are all finite!

S_t and R_t have well defined probability distributions that is dependent only on the preceding state and action: S_{t-1}, A_{t-1}. (Markov Property).

The "Markov" part in MDP comes from the Markov property.

Dynamics of the MDP: $p : S \times R \times S \times A \to [0, 1]$.

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

The function p defines the dynamics of the MDP, aka the transition function.

p: probability of the next state being s' and its reward being r, if we are in state s and take action a.

Reminder that p is a joint probability distribution for each choice of s and a (not r):

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

Additional Definitions:

State transition probabilities : $p : S \times S \times R \to [0, 1)$

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

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

Expected Rewards : $r : S \times A \to \mathbb{R}$

$$r(s, a) \doteq E\left[R_t \mid S_{t-1} = s,\ A_{t-1} = a \right]$$

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

Expected Reward for state-action-next state : $r : S \times A \times S \to \mathbb{R}$

$$r(s, a, s') \doteq E\left[R_t \mid S_{t-1} = s,\ A_{t-1} = a,\ S_t = s' \right]$$

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

3.2: Goals and Rewards

- At each time step t, the reward is a simple number $R_t \in \mathbb{R}$.
- The goal of the agent is to maximize cumulative reward in the long run.
- We can use the reward signal to give prior knowledge about how to achieve the task.

3.3: Returns and Episodes

G_t = Expected Return = expected sum of all future rewards

Episodic case:

- Simple definition of the expected return we want to maximize:

$$G_t = R_t + R_{t+1} + \dots + R_T$$ // sum of ALL future rewards.

- we get a reward at each time step, and T is the final time step.
- This approach makes sense when there is a notion of a final time step, that is, when the agent actions naturally break into sequences, that we call episodes.
- Each episode ends in a special state called the terminal state.
- It's important that the start of the episode is independent from the end of the previous episode.
- Tasks with episodes of this kind are called episodic tasks. In these, S is the set of all non-terminal states, and S^+ is the set of all states, including terminal ones.

Continuous States

- when the agent-environment interaction does not naturally break into episodes.
- Final timestep is ∞, so we can't really compute a useful G_t.
- we introduce discounting (to avoid a sum of returns going to ∞):

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

$$= \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$, where γ = discount rate, and

$0 \leq \gamma \leq 1$, (for $\gamma = 0$, the agent is myopic).

Recursive form of G_t

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

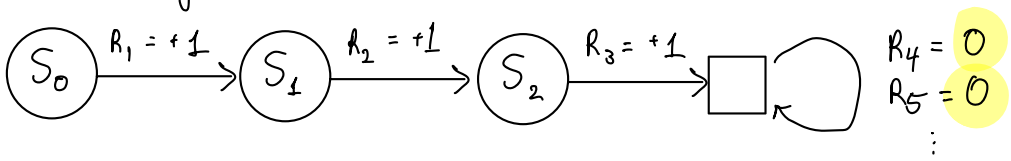
$$= R_{t+1} + \gamma \left(R_{t+2} + \gamma R_{t+3} + \dots \right)$$

$$= R_{t+1} + \gamma G_{t+1}$$

$$\Rightarrow \boxed{G_t = R_{t+1} + \gamma G_{t+1}}$$ // notice the recursion.

3.4 : Unified notation for Episodic and Continuing tasks

- S_t = state at timestep t (episodic case).
- The return over a finite # of terms of the episodic case can be treated as the infinite sum by adding an absorbing state:



unified notation for the return, that includes the possibility of $T = \infty$ or $\gamma = 1$:

$$G_t \doteq \sum_{k=t+1}^{T} \gamma^{k-t-1} R_k$$

// can be used for both episodic and continuing tasks.

3.5: Policies and value functions

Definitions:

- value function: expected value in a state (or a q-state for q-values)
- policy: mapping from states to probabilities of taking each possible action. Determines how the agent behaves.

The value function of a state s under a policy π, denoted as $v_\pi(s)$, is *defined* as the expected return (future rewards) by starting in s and following π thereafter:

$$\underbrace{v_\pi(s)}_{\text{value function of state } s \text{ following policy } \pi} \doteq \underbrace{E_\pi[G_t \mid S_t = s]}_{\text{Expected future rewards if the agent is in state } s \text{ and follows policy } \pi.} = E_\pi\left[\underbrace{\sum_{k=0}^{\infty} \gamma^k R_{t+k+1}}_{G_t = \text{sum of future rewards (discounted).}} \,\middle|\, S_t = s\right] \quad \forall s \in S$$

v_π = state-value function for policy π.

action-value function q_π:

$$q_\pi(s,a) \doteq \underbrace{E_\pi\left[G_t \mid S_t = s, A_t = a\right]}_{\text{expected future rewards if in state } s, \text{ take action } a, \text{ and then follow policy } \pi.} = E_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid S_t = s, A_t = a\right]$$

Difference btwn V_π and q_π is q_π conditions on $A_t = a$.

$V_\pi(s)$ and $q_\pi(s)$ can be determined by experience.
If we maintain an average reward received for each state (or q-state), this average converges to $V_\pi(s)$ (resp $q_\pi(s)$), as time passes.

Value functions also have a nice recursive property:

$$V_\pi(s) \doteq E_\pi[G_t \mid S_t = s]$$

// the value of state s under policy π is the expected return (cumulative discounted rewards) starting from state s

$$= E_\pi[\underbrace{R_{t+1}}_{\text{immediate reward at next timestep}} + \underbrace{\gamma G_{t+1}}_{\text{the discounted future reward from time } t+1 \text{ onward}} \mid S_t = s]$$

$$= \sum_a \underbrace{\pi(a \mid s)}_{\text{chance of action}} \sum_{s', r} \underbrace{p(s', r \mid s, a)}_{\text{chance of transition}} \underbrace{[r + \gamma E_\pi[G_{t+1} \mid S_{t+1} = s']}_{\text{value of transition}}$$

Sum over all available actions!

$\pi(a \mid s)$: probability of taking action a in state s under policy π

$p(s', r \mid s, a)$: probability of transitioning to state s' and receiving reward r given we're in state s and took action a

r: immediate reward of being in state s'

$E_\pi[G_{t+1} \mid S_{t+1} = s']$: expected future return starting from the next state s'

goodness of action.

$$= \boxed{\sum_a \pi(a \mid s) \sum_{s', r} p(s', r \mid s, a)[r + \gamma V_\pi(s')] \quad \forall s \in S}$$

// Key insight: $E_\pi[G_{t+1} \mid S_{t+1} = s'] \Leftrightarrow V_\pi(s')$

The future expected return from state s' is exactly the value function of s'!

It can be viewed as a sum on 3 values:

a, s', r.

For each triple, we compute the probability $\pi(a \mid s) p(s', r \mid s, a)$, weight the quantity in brackets by this probability, then sum over all the probabilities to get an expected value.

This is the **Bellman Eq for V_π**: it states a fundamental recursive relationship:

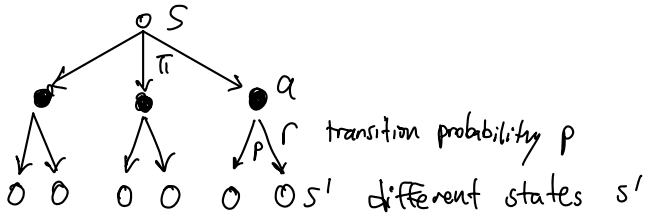
The value of being in a state =

immediate reward you can expect

⊕

discounted value of where you'll end up next.

We can represent this with a back-up diagram:



The value function V_π is the unique solution to its Bellman Eq and this forms the basis of a number of ways to compute, approximate, and learn V_π.

Eq connecting $V_\pi(s)$ with $q_\pi(s,a)$:

A state-value is an expectation over the available action-values under a policy:

$$V_\pi(s) \doteq E_{\pi,p}[G_t \mid S_t = s] \quad \text{// as before}$$

$$= E_{\pi,p}[q_\pi(s, A_t) \mid S_t = s]$$

$$= \sum_a \pi(a \mid s)\, q_\pi(s,a)$$

// this derivation relates $V_\pi(s)$ to $q_\pi(s,a)$. ■

The following recursive relationship statisfied by $V_\pi(s)$ is called the Bellman Eq for state-values.

The value function $V_\pi(s)$ is the unique solution to its Bellman Eq.

$$V_\pi(s) \doteq E_{\pi,p}\left[G_t \mid S_t = s\right]$$

$$= E_{\pi,p}\left[R_{t+1} + \gamma G_{t+1} \mid S_t = s\right]$$

$$= \sum_a \pi(a|s)\, E_{\pi,p}\left[R_{t+1} + \gamma G_{t+1} \mid S_t = s,\ A_t = a\right]$$

$$= \sum_a \pi(a|s) \sum_{s',r} p(s', r \mid s, a) \cdot E_{\pi,p}\left[r + \gamma G_{t+1} \mid S_t = s,\ A_t = a,\ S_{t+1} = s'\right]$$

$$= \sum_a \pi(a|s) \sum_{s',r} p(s', r \mid s, a) \cdot E_{\pi,p}\left[r + \gamma G_{t+1} \mid S_{t+1} = s'\right]$$

// by the Markov Property

$$= \sum_a \pi(a|s) \sum_{s',r} p(s', r \mid s, a) \cdot \left\{ r + \gamma E\left[G_{t+1} \mid S_{t+1} = s'\right] \right\}$$

$$= \sum_a \pi(a|s) \sum_{s',r} p(s', r \mid s, a) \left[r + \gamma V_\pi(s') \right]$$

// V to V relationship

In summary:

$$V_\pi(s) = \sum_a \left\{ \pi(a|s)\, q_\pi(s,a) \right\} \quad \text{(action-value perspective)}$$

$$= \sum_a \left\{ \pi(a|s) \sum_{s',r} \left[p(s', r \mid s, a)\left(r + \gamma V_\pi(s')\right) \right] \right\} \quad \text{(transition perspective)}$$

Bellman Eq for $q_\pi(s,a)$:

An action-value is an expectation over the possible next state-values under the environment's dynamics.

$$q_\pi(s,a) \doteq E_{p,\pi}\left[G_t \mid S_t = s,\ A_t = a\right]$$

- $q_\pi(s,a)$: action-value function for state s, taking action a, and then following policy π.
- $E_{p,\pi}[G_t]$: expected total return from time t onward
- $S_t = s,\ A_t = a$: given that we are in state s and take action a,

$$= E_{p,\pi}\left[R_{t+1} + \gamma G_{t+1} \mid S_t = s,\ A_t = a\right]$$

$$= E_{p,\pi}\left[R_{t+1} + \gamma v_\pi(S_{t+1}) \mid S_t = s,\ A_t = a\right]$$

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

// this derivation relates $q_\pi(s,a)$ to $v_\pi(s')$:

$q \rightarrow v$ relationship.

The following recursive relationship satisfied by the Bellman equation for action-values. The action-value function $q_\pi(s,a)$ is the unique solution to its Bellman Eq.

$$q_\pi(s,a) \doteq E_{p,\pi}\left[G_t \mid S_t = a,\ A_t = a\right] \quad \text{// as before}$$

$$= E_{p,\pi}\left[\underbrace{R_{t+1}}_{\text{immediate reward}} + \underbrace{\gamma G_{t+1}}_{\text{future reward, discounted}} \mid S_t = s,\ A_t = a\right]$$

total reward

$$= \sum_{s',r} p(s', r \mid s, a)\, E_{\pi,p}\left[r + \gamma G_{t+1} \mid S_t = s,\ A_t = a,\ S_{t+1} = s' \right]$$

$$= \sum_{s',r} p(s', r \mid s, a)\, E_{\pi,p}\left[r + \gamma G_{t+1} \mid S_{t+1} = s' \right] \quad \text{// Markov property}$$

$$= \sum_{s',r} p(s', r \mid s, a) \left[r + \gamma\, E_{\pi,p}\left[G_{t+1} \mid S_{t+1} = s' \right] \right]$$

$$= \sum_{s',r} p(s', r \mid s, a) \left[r + \gamma \sum_{a'} \pi(a' \mid s')\, E_{p,\pi}\left[G_{t+1} \mid S_{t+1} = s',\ A_{t+1} = a' \right] \right]$$

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

// this derives the $q \to q$ relationship.

In summary:

$$q_\pi(s, a) = \sum_{s',r} \left\{ p(s', r \mid s, a) \left[r + \gamma v_\pi(s') \right] \right\} \quad \text{// q-v}$$

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

// q-q

3.6 : Optimal Values and Optimal Policies

The Bellman equation shows us how to compute the value of a state for any policy π.

The Bellman optimality equations focus on optimal policy (the policy with the most reward over the long run).

Mathematically:

- $\pi \geq \pi'$ iif $v_\pi(s) \geq v_{\pi'}(s) \quad \forall s \in S$.
- the optimal policy is $\geq$ all other policies.
- the optimal functions are called v_* and q_*

$\Rightarrow$

Optimal Policy:

$$\pi_*(s) = \arg\max_a q_*(s,a) \quad \forall s \in S$$

optimal state-value function:

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

Optimal action value function:

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

For a state-action pair, the optimal function gives the expected return for taking action a in state s and thereafter following an optimal policy.

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

Bellman Optimality Equations

It expresses the fact that the value of a state under the optimal policy must equal the expected return for the best action from that state:

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

$$= \max_a \mathbb{E}_{q_*}\left[G_t \mid S_t = s, A_t = a\right]$$

$$= \max_a \mathbb{E}_{\pi_*}\left[R_{t+1} + \gamma G_{t+1} \mid S_t = s, A_t = a\right]$$

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

$$V_*(s) = \max_a \sum_{s',r} p(s',r \mid s,a)\left[r + \gamma v_*(s')\right]$$

↑ recursion!

The last 2 equations are the form of Bellman optimality eq for v_*.

For q_*, we have:

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

$$= \sum_{s',r} p(s',r \mid s,a)\left[r + \gamma \underbrace{\max_{a'} q_*(s',a')}_{\| \; v_*(s')}\right]$$

? q_* is "better" than v_*, because q_* is a 2-step planning process. We know the values as well as the values of the actions in the next state.

since $\pi_*(s) = \arg\max_a q_*(s,a)$

$$\Rightarrow \pi_*(s) = \arg\max_a \sum_{s',r} p(s',r \mid s,a)\left[r + \gamma \underbrace{\max_{a'} q_*(s',a')}_{\| \; v_*(s')}\right]$$

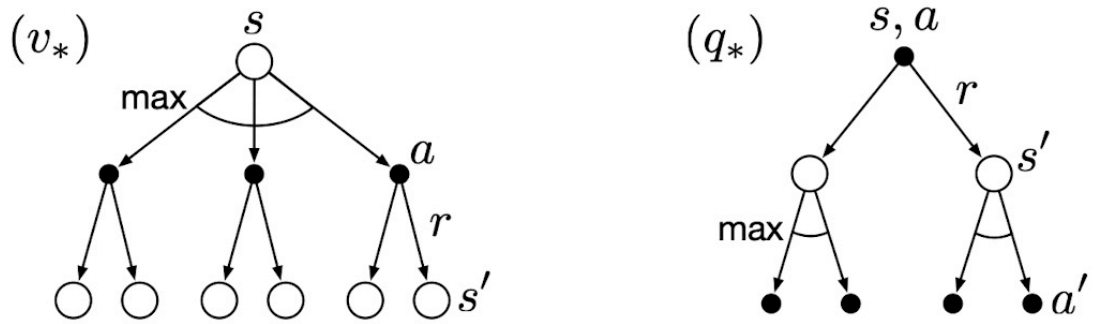


Figure 3.4: Backup diagrams for v_* and q_*

- for finite MDPs, the Bellman optimality Eq for v_* has a unique solution.
  It's a system of equations, one for each state (one unknown for each state).
- once we have v_*, it's easy to find the optimal policy (greedy wrt the optimal evaluation function v_*).

This solution relies on 3 assumptions which may not be true in reality:
1) we know the dynamics of the env.
2) we have enough computational resources.
3) Markov property.

3.7: Optimality and approximation

In a pratical setting, we almost always have to settle for an approximation. we can do this cleverly: make approximately optimal policies in states that have high probability of occuring, at the expense of making poor decisions in the states that have low probability of occuring.

3.8: Summary

The optimal value function is unique, the optimal policy isn't.

3.10: Partially Observable MDPs