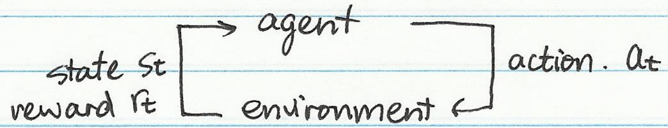
Reinforcement Learning

Q: How should decision-making agents act and learn from experience in the field?



Ex. robot navigating, game playing

* challenges
  - handling of uncertainty
  - exploration v.s. exploitation dilemma
  - delayed v.s. immediate reward? "temporal credit assignment"
  - evaluative feedback v.s. ~~con~~instructive feedback
  - complex worlds; computational guarantees.

Markov Decision Processes (MDPs)

* Definition
  - state space S with state $s \in S$
  - action space A with action $a \in A$
  - transition probabilities
    for all state-action pairs (s, a)

$$\mathbb{P}(s' \mid s, a) = \mathbb{P}(s_{t+1} = s' \mid s_t = s, a_t = a)$$

"probability of moving from state s to state s' after taking action a (at any time t)"

Assumptions:

- time independent

$$\mathbb{P}(s_{t+1} = s' \mid s_t = s, a_t = a) = \mathbb{P}(s_t = s' \mid s_{t-1} = s, a_{t-1} = a)$$

- Markov condition

$$\mathbb{P}(s_{t+1} \mid s_t, a_t) = \mathbb{P}(s_{t+1} \mid s_t, a_t, a_{t-1}, a_{t-2} \cdots)$$

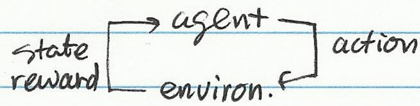
* Definition (cont.)
  - reward function

    $R(s, s', a)$ "real valued reward after taking action a in state s and moving to s'"

- simplifications for CSE 150.
  - Reward function $R(s, s', a) = R(s)$. "reward only depend on the current state".
  - Reward <u>bounded</u> and deterministic.
    i.e. $\max_s |R(s)| < \infty$
  - discrete, finite state space.
  - discrete, finite action space.

  } v.s. continuous, infinite.

Example: back-gammon.



S: board position. and agent's roll of dice

A: set of possible moves.

$$R(s) = \begin{cases} +1 & \text{win} \\ -1 & \text{lose} \\ 0 & \text{otherwise.} \end{cases}$$

$P(s' | s, a)$: how state changes due to agent's move, opponent rolls dice, opponent moves, agent rolls dice.

* Decision-making.

- policy: deterministic mapping from states to actions.
  $\pi: S \to A$
- # of policies: $\cancel{|A| \cdot |S|}$ $|A|^{|S|}$
- dynamics: $\mathbb{P}(S' | S, \pi(S))$
- experience under policy π.

  state $s_0 \xrightarrow[a_0 = \pi(s_0)]{\text{action}} s_1 \xrightarrow[a_1 = \pi(s_1)]{\text{action}} s_2 \rightarrow \cdots$

  reward. $r_0 \qquad r_1 \qquad r_2$

* How to measure accumulated reward over time?

- discount factor $0 \le \gamma \le 1$.

"long term discounted return"

$$= \sum_{t=0}^{\infty} \gamma^t \cdot r_t$$

possibilities:
- $\gamma = 0 \rightarrow$ only immediate reward at $t=0$ matters
- $\gamma \ll 1 \rightarrow$ near-sight agent
- $\gamma \approx 1 \rightarrow$ far-sighted agent

- intuitively: "near future is weighted more heavily than distant future."
- mathematically convenient, leads to recursive algorithms.

* State value function.

$V^{\pi}(s)$ = "expected discounted return following policy π from initial state s."

$$V^{\pi}(s) = \mathbb{E}^{\pi}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t) \,\middle|\, s_0 = s\right]$$

* Relating value function in different states

$$\begin{aligned} V^{\pi}(s) &= \mathbb{E}^{\pi}\left[R(s_0) + \gamma \cdot R(s_1) + \gamma^2 R(s_2) + \cdots \,\middle|\, s_0 = s\right] \\ &= R(s) + \gamma \cdot \mathbb{E}^{\pi}\left[R(s_1) + \gamma \cdot R(s_2) + \cdots \,\middle|\, s_0 = s\right] \\ &= R(s) + \gamma \sum_{s'} P(s' | s, \pi(s)) \cdot \mathbb{E}^{\pi}\left[R(s_1) + \gamma \cdot R(s_2) + \cdots \,\middle|\, s_1 = s'\right] \\ &= R(s) + \gamma \cdot \sum_{s'} P(s' | s, \pi(s)) \cdot V^{\pi}(s') \end{aligned}$$

$$V^{\pi}(s) = R(s) + \gamma \cdot \sum_{s'} P(s' | s, \pi(s)) \cdot V^{\pi}(s')$$ "Bellman equation"

* Optimality in MDPs.

Theorem: there is always at least one policy π^* ← optimal policy.
for which $V^{\pi^*}(s) \geq V^{\pi}(s)$. for all states and policies.

Goal: how to compute π^*? demo