Reinforcement Learning:

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

\[ \text{state } S_t \xrightarrow{\text{agent}} \text{action } A_t \xrightarrow{\text{environment}} \text{reward } R_t \]

Ex.: robot navigating, game playing

* challenges.
  - handling of uncertainty.
  - exploration vs. exploitation dilemma.
  - delayed vs. immediate reward? "temporal credit assignment"
  - evaluative feedback vs. constructive 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 \( (s, a) \)
    \[ P(s' \mid s, a) = P(S_{t+1} = s' \mid S_t = s, A_t = a) \]
    "probability of moving from state \( s \) to state \( s' \) after taking action \( a \) (at any time \( t \))"

Assumptions:
  - time independent
    \[ P(S_{t+1} = s' \mid S_t = s, A_t = a) = P(S_t = s' \mid S_{t-1} = s, A_{t-1} = a) \]
  - Markov condition.
    \[ P(S_{t+1} \mid S_t, A_t) = P(S_{t+1} \mid S_t, A_t, A_{t-1}, A_{t-2}, \ldots) \]

* 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 bounded and deterministic.
    i.e. \( \max_s |R(s)| < \infty \)
  - discrete, finite state space. \} v.s. continuous, infinite.
  - discrete, finite action space.

Example: backgammon.

\( S \): 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 die, opponent moves, agent rolls die.

**Decision-making.**

- policy: deterministic mapping from states to actions.
\( \pi : S \rightarrow A \)

- # of policies: \( |A| \)

- dynamics: \( \pi \)

- experience under policy \( \pi \).

State \( S_0 \) action \( S_1 \) action \( S_2 \) \( r_0 \) \( a_0 = \pi(S_0) \) \( r_2 \) \( a_1 = \pi(S_1) \) \( r_4 \) \( \rightarrow \ldots \)

**How to measure accumulated reward over time?**

- discount factor \( 0 \leq \gamma \leq 1 \).

"long term discounted return" possibilities:

\[
\sum_{t=0}^{\infty} \gamma^t r_t
\]

- \( \gamma = 0 \) \( \rightarrow \) only immediate reward at \( t=0 \) matters
- \( \gamma \ll 1 \) \( \rightarrow \) near-sight agent
- \( \gamma \approx 1 \) \( \rightarrow \) far-sight agent

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

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

\[ V^\pi(s) = \mathbb{E}^{\pi} \left[ \sum_{t=0}^{\infty} \gamma^t R(s_t) \mid s = s \right] \]

* Relating value function in different states

\[ V^\pi(s) = \mathbb{E}^{\pi} \left[ R(s_0) + \gamma R(s_1) + \gamma^2 R(s_2) + \cdots \mid s = s \right] \]

\[ = R(s) + \gamma \mathbb{E}^{\pi} \left[ R(s_1) + \gamma R(s_2) + \cdots \mid s = s \right] \]

\[ = R(s) + \gamma \sum_{s'} P(s' \mid s, \pi(s)) \cdot V^\pi(s') \]

\[ V^\pi(s) = R(s) + \gamma \sum_{s'} P(s' \mid s, \pi(s)) \cdot V^\pi(s') \quad \text{"Bellman equation"} \]

* Optimality in MDPs.

Theorem: there is always at least one policy \( \pi^* \) and \( \pi \) policy,

for which \( V^{\pi^*}(s) \geq V^\pi(s) \) for all states and policies.

Goal: how to compute \( \pi^* \)? (demo)
Review

* Reinforcement Learning

\[
\begin{align*}
\text{state } s_t \quad \xrightarrow{\text{agent}} \quad \text{action } a_t \\
\text{reward } r_t \quad \xrightarrow{\text{environment}} \\
\end{align*}
\]

* Markov Decision Process (MDP)

\[
\{S, A, P(s' \mid s, a), R(s)\}.
\]

States actions transition probabilities reward.

* Policy: assignment of states to actions \( \Pi(s) \in A \).

* State Value Function

\[
V^\Pi(s) = \mathbb{E}^\Pi \left[ \sum_{t=0}^{\infty} \gamma^t R(s_t) \mid s_0 = s \right] \quad \text{"expected value of discounted reward"}
\]

\[
V^\Pi(s) = R(s) + \gamma \sum_{s'} P(s' \mid s, \Pi(s)) \cdot V^\Pi(s'). \quad \text{Bellman Equation}
\]

* Action value Function

\[
Q^\Pi(s, a) = \quad \text{"expected return from initial state } s, \text{ taking action } a, \text{ then following policy } \Pi"
\]

\[
Q^\Pi(s, a) = \mathbb{E}^\Pi \left[ \sum_{t=0}^{\infty} \gamma^t R(s_t) \mid s_0 = s, a_0 = a \right]
\]

\[
Q^\Pi(s, a) = R(s) + \gamma \sum_{s'} P(s' \mid s, a) \cdot V^\Pi(s')
\]

* Optimality

Theorem: there is always at least one policy \( \Pi^* \) for which

\[
V^{\Pi^*}(s) \geq V^\Pi(s) \quad \text{for all } s, \Pi.
\]

* Optimal value functions

\[
V^*(s) = V^{\Pi^*}(s)
\]

\[
Q^*(s, a) = Q^{\Pi^*}(s, a) \quad \text{There may be many (equivalently) optimal policies, but optimal value functions (both state and action) are unique.}
\]
* Relations between value functions and policies

- Given MDP $\{S, A, P(s'|s,a), R(s), \gamma\}$ and
  
  \[ V^*(s) = R(s) + \gamma \sum_{s'} P(s'|s,a) \cdot V^*(s') \]

\[ Q^*(s,a) = R(s) + \gamma \sum_{s'} P(s'|s,a) \cdot V^*(s') \]

- Vice versa. (From optimal value functions to optimal policies)

\[ \pi^*(s) = \arg\max_a Q^*(s,a) \]

\[ = \arg\max_a \left[ R(s) + \gamma \sum_{s'} P(s'|s,a) \cdot V^*(s') \right] \]

\[ = \arg\max_a \left[ \sum_{s'} P(s'|s,a) \cdot V^*(s') \right] \]

* Planning under Uncertainty

Assume complete model of environment as.

MDP $\{S, A, P(s'|s,a), R(s), 0 \leq \gamma \leq 1\}$.

how to compute $\pi^*(s)$, or equivalently, $V^*(s)$ or $Q^*(s,a)$?

1) Policy Evaluation: how to compute $V^\pi(s)$ for any (possibly non-optimal) policy $\pi$?

From Bellman equation:

\[ V^\pi(s) = R(s) + \gamma \sum_{s'} P(s'|s,\pi(s)) \cdot V^\pi(s') \]

given

This is a system of $n$ linear equation for $n$ unknowns.

Put all unknowns on LHS.

\[ V^\pi(s) = \gamma \sum_{s'} P(s'|s,\pi(s)) \cdot V^\pi(s') = R(s) \]

\[ \sum_{s'} [I(s,s') - \gamma \sum_{s'} P(s'|s,\pi(s))] \cdot V^\pi(s') = R(s) \text{ for } s=1,2,\ldots,n. \]

Rewrite equation as: known $n \times 1$ vector.

\[ (I - \gamma P) V = R \]

$n \times n$ identity matrix unknown $n \times 1$ vector.
Solution: \[ V^\pi = (I - \gamma P^\pi)^{-1} R \]

- Matrix inversion is \(O(n^3)\) operation.

Example:
States \(S = \{0, 1\}\),
Transitions \(P^\pi(s', s, \pi(s)) = \begin{pmatrix} P_{00} & P_{01} \\ P_{10} & P_{11} \end{pmatrix}\) "rows sum to one",
Rewards \(R(s) = \begin{pmatrix} r_0 \\ r_1 \end{pmatrix}\)
State value function \(V^\pi(s) = \begin{pmatrix} V_0 \\ V_1 \end{pmatrix}\).
Solve:
\[
\begin{pmatrix}
1 & 0 \\
0 & 1
\end{pmatrix} - \gamma \begin{pmatrix} P_{00} & P_{01} \\ P_{10} & P_{11} \end{pmatrix} \begin{pmatrix} V_0 \\ V_1 \end{pmatrix} = \begin{pmatrix} r_0 \\ r_1 \end{pmatrix}
\]
2 equations for 2 unknowns.

2) Policy improvement.

* How to compute \(\pi'\) such that \(V^{\pi'}(s) \geq V^\pi(s)\) for all states \(s)?
* Recall \(Q^\pi(s, a)\) "expected return from state \(s\), follow action \(a\), then follow policy \(\pi\)."

How to compute \(Q^\pi(s, a)\)?
- Evaluate policy to get \(V^\pi(s)\).
- \(Q^\pi(s, a) = R(s) + \gamma \sum_{s'} P(s' | s, a) \cdot V^\pi(s')\)

Define "greedy policy":
\[
\pi'(s) = \arg\max_a Q^\pi(s, a)
= \arg\max_a \left[ \sum_{s'} P(s' | s, a) \cdot V^\pi(s') \right]
\]

Theorem: greedy policy \(\pi'\) everywhere performs better or equal to original policy \(\pi\).
\(V^{\pi'}(s) \geq V^\pi(s)\) for all \(s\).

Intuition: if better to choose action \(a\) in state \(s\), then follow \(\pi'\), it's always better to choose action \(a\) in state \(s\).