Reinforcement Learning

RL vs supervised learning

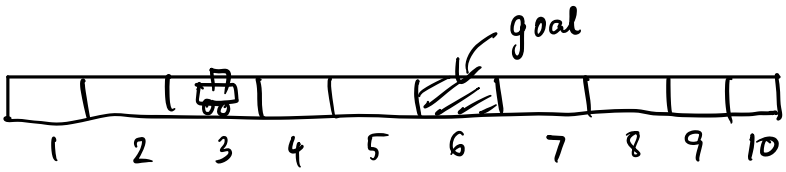
① sequential decision making
- long term future consequences
- greedy is not optimal
- return vs. risk tradeoffs

② No or little human supervision

③ learn from designed reward function

④ Can collect more data iteratively

Markov Decision Process



S : set of states (all possible configurations)

A : set of actions ($A = \{L, R\}$)

P_{sa} : dynamics / transitions
state transition prob. distrib"
at state, when applying action a
the dist" of next s'

$P_{sa}(s') : Pr[s' \mid s, a]$

P_{sa} is a distrib"

$P_{sa} \in \mathbb{R}^{|S|}$ $\quad \sum_{s' \in S} P_{sa}(s') = 1$

Example: L action succeeds w. prob 0.9

$P_{7,L}(6) = 0.9$

$P_{7,L}(7) = 0.1$

$P_{7,L}(s') = 0 \quad \forall\, s' \neq 6, 7$

R action succeeds w. prob. 0.8

$P_{7,R}(8) = 0.8 \quad P_{7,R}(7) = 0.2$

$P_{7,R}(s') = 0 \quad \forall\ s' \neq 7, 8$

Once you reach 6, robot stays there:

$P_{6,L}(6) = 1, \quad P_{6,L}(s') = 0 \quad \forall\ s' \neq 6$

$P_{6,R}(6) = 1, \quad \ldots$

Sequential decision making process

- s_0 initial state (given)
- algo chooses action $a_0 \in A$
- environment step: $s_1 \sim P_{s_0 a_0}$
- algo sees s_1, takes action a_1
- environment: $s_2 \sim P_{s_1, a_1}$

POMDP (partially observed MDP)

$s_0 = 4$
$a_0 = R$
$s_1 \sim P_{4,R}$
$s_1 = 5$
$a_1 = R$
⋮

Reward function

$R: S \to \mathbb{R}$
$R(s)$

$R(6) = 1.0$
$R(s) = -0.1 \quad \forall\ s \neq 6$

$s_0 \xrightarrow{a_0} s_1 \xrightarrow{a_1} s_2 \xrightarrow{a_2} \ldots$ trajectory / episode

Total payoff:

$R(s_0) + R(s_1) + \ldots R(s_t) + \ldots$

Discounted total payoff (infinite horizon)

Discount factor $\gamma < 1$ (interest rate)

$R(s_0) + \gamma R(s_1) + \ldots + \gamma^t R(s_t) + \ldots$

Discounted reward is bounded

Suppose $R(s) \in [-M, M]$

$\Rightarrow$ total payoff $\leq M + \gamma M + \gamma^2 M + \ldots$

$$-\frac{M}{1-\gamma} \leq \qquad = \frac{M}{1-\gamma}$$

finite horizon : stop process at time T

Expected payoff :

$\mathbb{E}[R(s_0) + \gamma R(s_1) + \ldots + \gamma^t R(s_t) + \ldots]$

Reward can be $R(s, a)$ ← function of state & action

$MDP(S, A, \{P_{sa}\}_{\substack{s \in S \\ a \in A}}, \gamma, R)$

goal: maximize expected discounted payoff.

Markov property :

at any time t, given s_t, all future states s_{t+j} doesn't depend on past states/actions $s_0, a_0, \ldots s_{t-1}, a_{t-1}$

$\Rightarrow$ optimal action at time t does not depend on past states and actions

$\Rightarrow$ best strategy can be expressed as a policy

$\pi : S \longrightarrow A$ policy

$a_t = \pi(s_t)$

$\pi(s) = R$ if $s \leq 6$

$\pi(s) = L$ if $s \geq 7$

randomized policy $\pi(a|s)$

$\exists$ deterministic optimal policy

Value function:

$V^\pi : S \to \mathbb{R}$

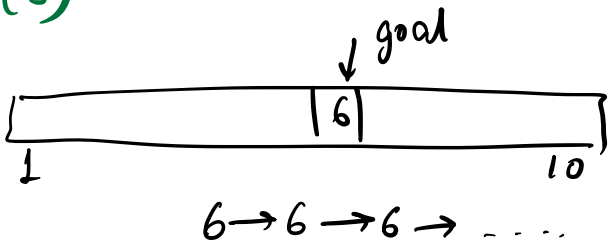
$V^\pi(s) \triangleq$ Expected total payoff of executing policy π starting from state s

$s_0, a_0, \ldots$

$a_t = \pi(s_t)$

$$V^\pi(s) = \mathbb{E}\left[R(s_0) + \gamma R(s_1) + \ldots + \gamma^t R(s_t) + \ldots \mid s_0 = s\right]$$

How to compute $V^\pi(s)$



$6 \to 6 \to 6 \to \ldots$

$$V^\pi(6) = 1 + \gamma + \gamma^2 + \ldots \gamma^t + \ldots$$

$$= \frac{1}{1-\gamma}$$

$$V^\pi(6) = 1 + \gamma \underbrace{\left(1 + \gamma + \gamma^2 + \ldots\right)}_{V^\pi(6)}$$

$$V^\pi(6) = 1 + \gamma V^\pi(6)$$

$$V^\pi(6) = \frac{1}{1-\gamma}$$

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

$$= R(s) + \gamma \underbrace{\mathbb{E}\left[R(s_1) + \gamma R(s_2) \ldots \mid s_0 = s\right]}_{\text{total payoff starting from } s_1 \text{ and executing } \pi}$$

$$= R(s) + \gamma \mathop{\mathbb{E}}_{s_1 \sim P_{s,\pi(s)}}\left[R(s_1) + \gamma R(s_2) + \ldots\right]$$

$$= R(s) + \gamma \mathop{\mathbb{E}}_{s_1 \sim P_{s,\pi(s)}}\left[V^\pi(s_1)\right]$$

$$= R(s) + \gamma \sum_{s' \in S} P_{s,\pi(s)}(s') \cdot V^\pi(s')$$

$$V_\pi(s) = R(s) + \gamma \sum_{s' \in S} P_{s,\pi(s)}(s') \cdot V^\pi(s')$$ Bellman Equation

$V_\pi(s)$ as variables

$|S|$ variables, $|S|$ equations

Solving linear system of eqn $\Rightarrow V^\pi(s), \forall s$

given π, $P_{s,a}$ $\xrightarrow[\text{linear system}]{\text{solve}}$ $V^\pi(s), \forall s$

Next: $V^*(s) = \max_\pi V^\pi(s)$

$\pi^* = \arg\max_\pi V^\pi(s)$

~~$\pi^*(s)$~~

Markov Property $\Rightarrow \exists \pi^*$ that simultaneously maximizes $V^\pi(s) \ \forall \ s \in S$

$\pi^* : S \longrightarrow A$

Instead of computing π^* directly

let's calculate $V^*(s) \triangleq V^{\pi^*}(s)$

We will get π^* from $V^*(s)$

Bellman Equation for V^*

$$V^*(s) = \max_\pi V^\pi(s)$$

$$= \max_\pi \left(R(s) + \gamma \sum_{s'} P_{s\,\pi(s)}(s') \cdot V^\pi(s') \right)$$

$$= R(s) + \max_\pi \left(\gamma \sum_{s'} P_{s,\pi(s)}(s') \cdot V^\pi(s') \right)$$

$$= R(s) + \max_a \left(\gamma \sum_{s'} P_{s,a}(s') \cdot \max_\pi V^\pi(s') \right)$$

$$= R(s) + \gamma \max_a \sum_{s'} P_{s,a}(s') \cdot V^*(s')$$

$$V^*(s) = R(s) + \gamma \max_a \sum_{s'} P_{s,a}(s') \cdot V^*(s')$$

(new on $V^*(s)$; old on $V^*(s')$)

Bellman Operator

best expected payoff after taking action a

$V^* \in \mathbb{R}^{|S|}$ $\quad \begin{bmatrix} V^*(1) \\ \vdots \\ V^*(|S|) \end{bmatrix}$

B operator $\quad \mathbb{R}^{|S|} \longrightarrow \mathbb{R}^{|S|}$

$V^* = B(V^*)$

Value Iteration

Initialize $V \in \mathbb{R}^{|S|}$, $V = 0$, $V(s) = 0 \ \forall s \in S$

Loop:

$V := B(V)$

$\forall s \in S, \quad V(s) = R(s) + \gamma \max_a \sum_{s'} P_{s,a}(s') \cdot V(s')$

($V(s)$: new; $V(s')$: old)

$V \longrightarrow V^*$ because B is a contraction

$$\pi^*(s) = \arg\max_a \sum_{s' \in S} P_{s,a}(s') \cdot V^*(s')$$

Policy Iteration

Initialize a policy π

- Loop {
  - let $V = V^\pi$ (obtained by solving system of eqn)
  - $\pi(s) = \arg\max_a \sum_{s'} P_{sa}(s') \cdot V(s')$

}

V^*, π^* is stationary point of this algorithm

Policy Iteration

Con: each iteration needs solving linear system

Pro: converge in finite time complexity

A is finite

S is finite

Atmost $|A|^{|S|}$ policies

Practice: discrete state space

- Value iteration often slightly better
- continuous controls: policy iteration