Reinforcement learning

Reinforcement learning - decision making
- No supervisor
- No instantaneous feedback
- Agent gets action → influence environment
  ∴ influencing the data it can see

The RL Problem:-

- R_t → Agent work is to maximise cumulative reward

Reward Hypothesis:-

All goals ⟷ maximization of expected cumulative reward.

:- Sequential decision making → goal is long term.

Time series of A_t, O_t, R_t → RL

$H_t = A_1, O_1, R_1, \ldots, A_t, O_t, R_t$

Now algorithm → mapping from History to action

State → info to determine what happens next

$$S_t = f(H_t)$$

S_t^e → info/numbers to pick next obs^n/reward
- ↳ not seen by agent

S_t^a → info/numbers to pick next action,
- ↳ used by RL algos.

$$S_t^a = f(H_t)$$

Information State:

Markov property

$$P[S_{t+1} \mid S_t] = P[S_{t+1} \mid S_1, \ldots, S_t]$$

$H_{1:t} \rightarrow (S_t) \rightarrow H_{t+1:\infty}$

(future is independent of past), given the present is enough to determine future

$\therefore S_t^e \rightarrow$ Markov & H_t is markov (By def.)

env. state actually uses this state to determine future actions = Markov

What happens next - depends on kind of state that we pick as well!!! (Rat example)

Fully observable: $O_t = S_t^a = S_t^e$

↳ Markov Decision Process (MDP)

Partially obs. state: agent state ≠ env. state

Beliefs: $S_t^a = \begin{pmatrix} P(S_t^e = s^1) & \ldots & P(S_t^e = s^n) \end{pmatrix}$

keeping prob distribution

or RNN: $S_t^a = \sigma(S_{t-1}^a W_s + O_t W_o)$

RL agent could include following.

Policy: agent's behaviour fn:
↳ how agent picks its actions

value fn: how good to be at particular state/action
↳ how much reward to expect.

Model: How agent thinks environment works

Policy: Deterministic policy $a = \pi(s)$

Stochastic policy $\pi(a|s) = P(A=a|S=s)$

Value: ⇒ goodness / Badness of state.

$$v_\pi(s) = E_\pi(R_t + \gamma R_{t+1} \dots | S_t = s)$$

↓
Behaviour that maximises this → less enough risk.

Model: Transition → dynamics
Rewards → Immediate reward.

$$P^a_{ss'} = P(S'=s' | S=s, A=a)$$

$$R^a_s = E(R | S=s, A=a)$$ ← overall prediction of reward.

Value Based:
value fn included
⇒ No policy fn (Implicit understood)

How RL and planning problems are different

↓
need to balance Exploration & Exploitation

Find more info maybe by losing reward → Given info - maximise reward

Lecture 2 :

Bandits — MDPs with 1 state!

$$P_{ss'} = P[S_{t+1} = s' / S_t = s]$$

↓

Put '1 to n' to make a matrix

$$P = \text{from} \begin{bmatrix} P_{11} & \cdots & P_{1n} \\ \vdots & & \\ P_{n1} & & P_{nn} \end{bmatrix}$$

to

from to

Markov process (Markov chain) $\langle S, P \rangle$

S = {set of all possible states}

P = {transition matrix}

Sample = random sequence of these dynamics

Markov reward process

Reward fn $R_s = E(R_{t+1} / S_t = s)$

↑

Just the immediate reward!

return G_t → maximize goal / return

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

↓

G is a random sample ⇒ So no expectation

γ close to '0' → myopic evaluation

γ close to 1 → far sighted evaluation

discount factor → uncertainty about rewards in future

Value for:-

$$V(s) = E(G_t \mid s_t = s)$$

how much reward will you get until you terminate!

Bellman eqn:-

$$V(s) = E(R_{t+1} + \gamma G_{t+1} \mid s_t = s)$$

$$= E(R_{t+1} + \gamma V(s_{t+1}) \mid s_t = s)$$

By law of expectation of expectation

$$V(s) = E(R_{t+1} + \gamma V(s_{t+1}) \mid s_t = s)$$

$$V(s) = R_s + \gamma \sum_{s' \in S} P_{ss'} V(s')$$

$$V = R + \gamma P V \rightarrow \text{vector rep}^n$$

$$V = (I - \gamma P)^{-1} R$$

another level of complexity :-

$$P_{ss'}^{a} = P[S_{t+1}=s' \mid S_t=s, A=a]$$

$$\pi(a|s) = P(A_t=a \mid S_t=s)$$

↑ This is what agent controls !!!

Policies.

MDP & MRP.

$S_1\ S_2 \cdots$ is Markov process $\langle S, P^{\pi} \rangle$

$S_1, R_2\ S_2 \cdots$ is mrk. Rard proces

$\langle S, P^{\pi}, R^{\pi}, \gamma \rangle$

where $P^{\pi}_{s,s'} = \sum_{a \in A} \pi(a|s)\, P^{a}_{ss'}$

$$R_s^{\pi} = \sum_{a \in A} \pi(a|s)\, R_s^{a}$$

action-value fn :-

$$q_{\pi}(s,a) = E_{\pi}(G_t \mid S_t=s, A_t=a)$$

$$q_{\pi}(s,a) = E_{\pi}(R_{t+1} + \gamma q_{\pi}(S_{t+1}, A_{t+1}) \mid S_t=s, A_t=a)$$

$$v_\pi(s) = \sum_{a \in A} \pi(a|s)\, q_\pi(s,a)$$

$$q_\pi(s,a) = R_s^a + \gamma \sum_{s' \in S} P_{ss'}^a v_\pi(s')$$

$$\Downarrow$$

$$v(s) = \sum_{a \in A} \pi(a|s) \left(R_s^a + \gamma \sum_{s' \in S} P_{ss'}^a v_\pi(s') \right)$$

$$q_\pi(s,a) = R_s^a + \gamma \sum_{s' \in S} P_{ss'}^a \sum_{a' \in A} \pi(a'|s')\, q_\pi(s',a')$$

$$v_\pi = R^\pi + \gamma P^\pi v_\pi$$

$$v_\pi = (I - \gamma P^\pi)^{-1} R^\pi$$

Optimal value fn:-

$$v_*(s) = \max_\pi v_\pi(s)$$

$$q_*(s,a) = \max_\pi q_\pi(s,a)$$

↙

given action, over all policies

Solving MDP - finding q_*

Optimal policy:-

$\pi \geq \pi'$ if $v_\pi(s) \geq v_{\pi'}(s)$; $\forall s$

Theorem: ∃ an optimal policy s.t

$\forall \pi \geq \pi \quad \forall \pi$

$v_{\pi_*}(s) = v_*(s)$

$q_{\pi_*}(s,a) = q_*(s,a)$

Deterministic optimal policy

$$\pi_*(a|s) = \begin{cases} 1 & \text{if } a = \underset{a \in A}{\operatorname{argmax}}\, q_*(s,a) \\ 0 & \text{otherwise} \end{cases}$$

Bellman optimality eqn:

$$v_*(s) = \max_a q_*(s,a)$$

$$q_*(s,a) = R_s^a + \gamma \sum_{s' \in S} P_{ss'}^a v_*(s')$$

$$v_*(s) = \max_a \left(R_s^a + \gamma \sum_{s' \in S} P_{ss'}^a v_*(s') \right)$$

$$q_*(s,a) = R_s^a + \gamma \sum_{s' \in S} P_{ss'}^a \max_{a'} q_*(s',a')$$

Lecture 3:

Bellman eqn → recursive decomposition

value fn → stores / cached information

Planning in MDP:

For prediction

Input: MDP or MRP with π output V_π

For control

Input: MDP output V_* & π_*

Iterative Policy evaluation:

Synchronous backup:

~~Update~~ ~~#iteration k+1~~ ~~$(\forall s \in S)$~~ ~~$V_{k+1}(s)$~~ for update

For all iteration, for all 's', update $V_{k+1}(s)$ from $V_k(s')$

~~s' → Successor state of s~~

Bellman Expectation equation:

$$V^{k+1} = R^\pi + \gamma P^\pi V^k$$

Iterative policy:

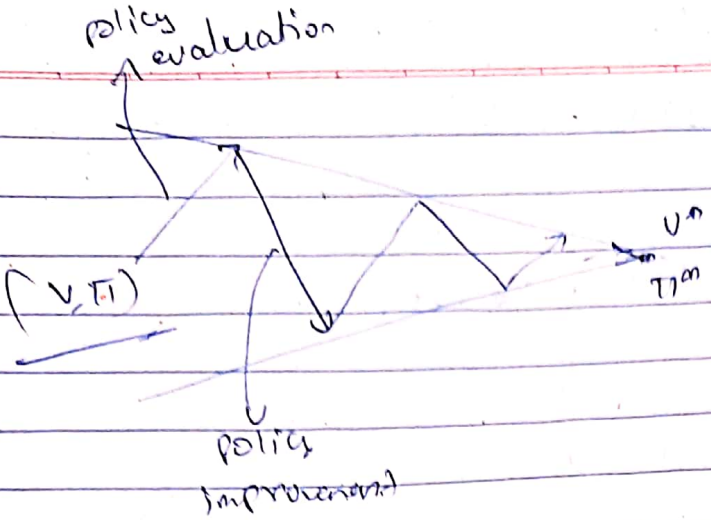
V_k → Random policy → make actions not randomly But using these policy.

Value fn → figure out optimal policy

Policy iteration:

Evaluate policy — $V_\pi(s) = \dots$

Improve policy — choosing better action. i.e greedy(V_π)



~~Ch~~ $s \in S$ $\quad a = \pi(s)$

$$\pi'(s) = \underset{a \in A}{\arg\max}\ q_{\pi}(s,a)$$

$$q_{\pi}(s, \pi'(s)) = \max_{a \in A} q_{\pi}(s,a) \geq q_{\pi}(s, \pi(s)) = v_{\pi}(s)$$

~~$\therefore v_{\pi'}(s) \geq v_{\pi}(s)$~~

$$v_{\pi} \leq q_{\pi}(s, \pi'(s))$$

$$= E_{\pi'}\left[R_{t+1} + \gamma v_{\pi}(S_{t+1}) \mid S_t = s\right]$$

$$\leq E_{\pi'}\left(R_{t+1} + \gamma q_{\pi}(S_{t+1}, \pi'(S_{t+1})) \mid S_t = s\right]$$

$$\vdots$$

$$\leq E_{\pi'}\left[R_{t+1} + \gamma R_{t+2} + \cdots \mid S_t = s\right]$$

$$= v_{\pi'}(s)$$

If improvements stop..

$$q_{\pi}(s, \pi'(s)) = \max_{a \in A} q_{\pi}(s,a) = q_{\pi}(s, \pi(s))$$

$$= v_{\pi}(s)$$

equality of Bellman optimality eqn

$$\Rightarrow v_{\pi}(s) = v_{*}(s)$$

$\Rightarrow \pi$ is optimal policy..

Modified policy iteration:-

- ~~It~~ improve only
- Considers only few iterations

Value iteration:-

Principle of optimality:-

- Policy $\pi(a/s)$ achieves optimal value $V_\pi(s) = V_{opt}(s)$ iff
  $\forall$ s' reachable from s
  π achieves the optimal value from s'
  $V_\pi(s') = V_{opt}(s')$

$$V_{opt}(s) \leftarrow \max_{a \in A} \left(R_s^a + \gamma \sum_{s' \in S} P_{ss'}^a V_{opt}(s') \right)$$

Use Bellmann optimality eqn as backup.

$$V_{k+1} = \max_{a \in A} \left(R^a + \gamma P^a V_k \right)$$

- state-value fns $V_\pi(s)$ or $V_{opt}(s)$
  → complexity $O(mn^2)$ per iteration
  m actions & n states
- action-value function $q_\pi(s,a)$ or $q_{opt}(s,a)$
  → complexity $O(m^2n^2)$ per iteration

Asynchronous Dynamic programming:-

In Place Dyn. Prog:-

Free

Lecture 4 : Model Predictive control :-

Monte-Carlo learning :

value = mean return

Caveat : All episodes should terminate

$$V_\pi = E_\pi \left(G_t \mid S_t = s \right)$$

↑ used empirical mean

↳ terminates at a point!! Not at 'oo'

Two types :-

(i) First visit (ii) Every visit

For each episode :-

$S_0, A_0, R_1 \dots S_{T-1}, A_{T-1}, R_T$

→ visit all / first state

$N(s) \leftarrow N(s) + 1$

$S(s) \leftarrow S(s) + G_t$

$V(s) = S(s)/N(s)$

By central limit thm, as $N(s) \to \infty$ $V(s) \to V_\pi(s)$

Incremental mean :

$$\mu_k = \mu_{k-1} + \frac{1}{k}\left(x_k - \mu_{k-1}\right)$$

↑ defn of mean!

Incremental monte-carlo :

$N(S_t) \leftarrow N(S_t) + 1$

$$V(S_t) \leftarrow V(S_t) + \frac{1}{N(S_t)}\left(G_t - V(S_t)\right)$$

Sometimes we can use α

Temporal difference learning :

guess → guess → ...

TD(0) algorithm :

update value $V(S_t)$ toward estimated return

estimated return $= R_{t+1} + \gamma V(S_{t+1})$

$\therefore V(S_t) \leftarrow V(S_t) + \alpha \underbrace{(R_{t+1} + \gamma V(S_{t+1}) - V(S_t))}_{\delta}$

δ_t ← TD target → δ_t = TD error

Bias / Variance Trade-Off :

Return $G_t = R_{t+1} + \ldots + \gamma^{T-1} R_T$ ⇒ unbiased estimate of $V_\pi(S_t)$

$R_{t+1} + \gamma V_\pi(S_{t+1})$ = True target = unbiased estimate of $V_\pi(S_t)$

$R_{t+1} + \gamma V(S_{t+1})$ → biased estimate of $V_\pi(S_t)$

→ can be used in Random walk example.

Certainty Equivalence :

$$\sum_{k=1}^{K} \sum_{t=1}^{T_k} \left(G_t^k - V(S_t^k)\right)^2$$

Solution to MDP $(S, A, \hat{P}, \hat{R}, \gamma)$ that best fits data

$$\hat{P}_{ss'}^{a} = \frac{1}{N(s,a)} \sum_{k=1}^{K} \sum_{t=1}^{T_k} 1(s_t^k, a_t^k, s_{t+1}^k = s, a, s')$$

$$\hat{R}_s^a = \frac{1}{N(s,a)} \sum_{k=1}^{K} \sum_{t=1}^{T_k} 1(s_t^k, a_t^k = s, a)\, r_t^k$$

n step TD learning :

$n=1$ (TD) $G_t^{(1)} = R_{t+1} + \gamma V(S_{t+1})$

$n=2$ & $G_t^{(2)} = R_{t+1} + \gamma R_{t+2} + \gamma^2 V(S_{t+2})$

$n=\infty$ (MC) $G_t^{(\infty)} = R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^{T-1} R_T$

Suppose

$G_t^{(n)} = R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^{n-1} R_{t+n} + \gamma^n V(S_{t+n})$

n-step temporal difference :

$$V(S_t) \leftarrow V(S_t) + \alpha \left(G_t^{(n)} - V(S_t) \right)$$

λ-return :

$$G_t^{\lambda} = (1-\lambda) \sum_{n=1}^{\infty} \lambda^{n-1} G_t^{(n)}$$ weights $(1-\lambda)\lambda^{n-1}$

Forward-view TD(λ)

$$V(S_t) \leftarrow V(S_t) + \alpha \left(G_t^{\lambda} - V(S_t) \right)$$

Thm :

Sum of offline updates is identical for forward-view & backward view TD(λ)

$$\sum_{t=1}^{T} \alpha \delta_t E_t(s) = \sum_{t=1}^{T} \alpha \left(G_t^{\lambda} - V(S_t) \right) \mathbb{1}(S_t = s)$$

TD(1): $E_t(s) = \gamma E_{t-1}(s) + 1(S_t = s)$

Total error by end of episode

$= \delta_t = \gamma\delta_{t+1} + \gamma^2\delta_{t+2} \cdots \gamma^{T-1-t}\delta_{T-1}$

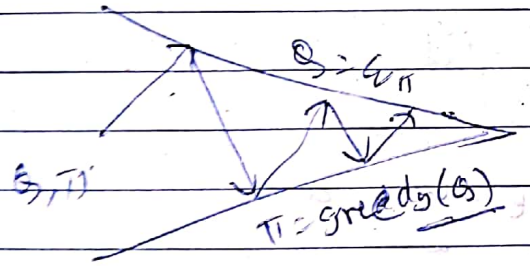
$\lambda = 1$: Sum of TD errors

$= G_t - V(S_t)$

For λ → $\delta_t + (\gamma\lambda)\delta_{t+1} + (\gamma\lambda)^2\delta_{t+2} + \cdots$

$$\sum_{t=1}^{T} \alpha\delta_t E_t(s) = \alpha \sum_{t=k}^{T} (\gamma\lambda)^{t-k}\delta_t = \alpha\left(G_k^{\lambda} - V(S_k)\right)$$

Lecture 5 Model-Free Control



ϵ Greedy exploration:

$$\pi(a/s) = \begin{cases} \frac{\epsilon}{m} + 1 - \epsilon & \text{if } a^* = \underset{a \in A}{\arg\max}\, Q(s,a) \\ \frac{\epsilon}{m} & \text{otherwise} \end{cases}$$

Thm:-

For any ϵ-greedy policy π, the ϵ-greedy policy π' with respect to q_π is improved

$$V_{\pi'}(s) \geq V_\pi(s)$$

$$q_{\pi}(s, \pi'(s)) = \sum_{a \in A} \pi'(a|s)\, q_{\pi}(s,a)$$

$$= \frac{\epsilon}{m} \sum_{a \in A} q_{\pi}(s,a) + (1-\epsilon) \max_{a \in A} q_{\pi}(s,a)$$

$$\geq V_{\pi}(s)$$

Greedy in the limit with infinite exploration (GLIE)

- $\lim_{k \to \infty} N_k(s,a) = \infty$
- $\lim_{k \to \infty} \pi_k(a|s) = 1 \left(a = \operatorname{argmax}_{a' \in A} Q_k(s,a')\right)$

GLIE monte Carlo ; $Q(s,a) \longrightarrow q_{\pi}(s,a)$

For sarsa algorithm :

$Q(s,a) \longrightarrow q_{\pi}(s,a)$ if

$\pi_t(a|s)$ $\sum_{t=1}^{\infty} \alpha_t = \infty$ and

$\sum_{t=1}^{\infty} \alpha_t^2 < \infty$

n Step Sarsa :

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \left(q_t^{(n)} - Q(S_t, A_t)\right)$$

$$q_t^{\lambda} = (1-\lambda) \sum_{n=1}^{\infty} \lambda^{n-1} q_t^{(n)}$$

$$\therefore Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \left(q_t^{\lambda} - Q(S_t, A_t)\right)$$

Off-policy learning :

$\{S_1, A_1, R_2, \ldots S_t\} \sim \mu$

$$E_{x \sim P}(f(x)) = \sum P(x) f(x)$$

$$= \sum Q(x) \frac{P(x)}{Q(x)} f(x)$$

$$= E_{x \sim Q}\left(\frac{P(x)}{Q(x)} f(x)\right)$$

$$G_t^{\pi/\mu} = \frac{\pi(A_t|S_t)}{\mu(A_t|S_t)} \frac{\pi(A_{t+1}|S_{t+1})}{\mu(A_{t+1}|S_{t+1})} \cdots \frac{\pi(A_t|S_t)}{\pi(A_t|S_t)} G_t$$

$$V(S_t) \leftarrow V(S_t) + \alpha\left(G_t^{\pi/\mu} - V(S_t)\right)$$

Off-policy control

$$\pi(S_{t+1}) = \underset{a'}{\operatorname{argmax}}\, Q(S_{t+1}, a')$$

Q learning target $= R_{t+1} + \max_{a'} \gamma Q(S_{t+1}, a')$