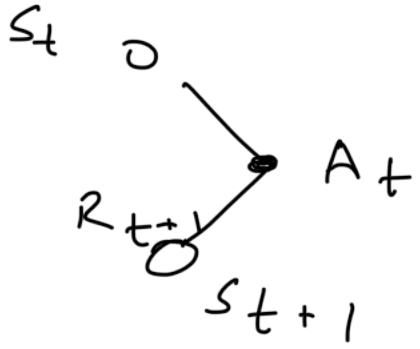
RL - Week-5

L1 TD (Temporal Difference) Learning

simplest
TD Method



estimate of $V(S_t)$ at time $(t+1)$

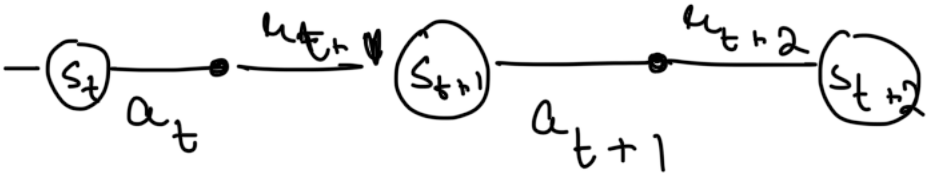
$$V(S_t) \leftarrow V(S_t) + \alpha \left[\underbrace{R_{t+1} + \gamma V(S_{t+1})}_{\text{estimate of } V(S_t) \text{ at time } (t+1)} - \underbrace{V(S_t)}_{V(S_t) \text{ at time } t} \right]$$

- explains complex behaviour, at the same time simple
- predic^n at time $t+1$ is better than predict^n at time t. So, we use future prediction to adjust the past predic^n. TD error (δ)
- Helps in behavioural psychology & neuroscience.
- not req. a complete model (like MC) only sampling.
- it fully bootstrapped. Learn before the final outcome. & learn w/o final outcome.

- TD advantage of both DP & MC.
Eg. Loan repayment data
- TD Predic^n (Evaluat^n)

TD(0): $V_{k+1}(s_t) \leftarrow V_k(s_t) + \alpha\left[r_{t+1} + \gamma V_k(s_{t+1}) - V_k(s_t)\right]$

no knowledge of probab. & reward but has a access to the sampled model.



ac^{ns} are according to the policy π.

$V(A) = 0.5$

$V(B) = 0.8$

r, $A \to B = 0$

$\alpha = 0.2$

$\gamma = 0.9$

$$V(A) = V(A) + \alpha\left[r + \gamma V(B) - V(A)\right]$$

$$= 0.5 + 0.2\left[\cancel{0} + (0.9 \times 0.8) - 0.5\right]$$

$\cancel{0.72 - 0.5}$

0.22

0.044

$$= 0.544$$

L2 MC Vs TD Updates

same set of trajectories

batch-TD updates uses episodes

A, 0, B, 0 → T

B, 1 → T

B, 0 → T

A, 1, B, 0 → T

B, 1 → T

B, 0 → T

$\gamma = 1$

because all trajectory terminate

MC

$$v(A) = \frac{0+1}{2} = 0.5$$

$$v(B) = \frac{1+0+0+1+0}{6} = \frac{\cancel{2}}{\cancel{6}} = \frac{1}{3}$$

TD

$$v(A) = E[R + \gamma V(B)]$$

$$= 0.5 + \gamma V(B)$$

$$= 0.5 + 0.33 = 0.83$$

forms an MDP.

$$V(B) = \frac{1}{3}$$

after B we always terminate

V(A)s are different.

- If ∞ samples are taken, then they both converge to same value.
- Otherwise, MC returns least square estimate of return. & TD returns certainty equivalency estimate.

L5 Sarsa

- TD Control $\rightarrow$ learns optimal policies.
  ↳ approximates optimal policy

GPI

$$\begin{tikzcd} \pi \arrow[r, bend left, "\text{evaluation}", "Q \to Q^{\pi}"'] & Q \arrow[l, bend left, "\pi \to \text{greedy}(Q)", "\text{improve}"'] \end{tikzcd}$$

→ policy evaluate by TD(0) on value func^n.

→ policy improvement → greedy approach wrt current value func^n.

→ we estimate ac^n values and not state values, iff. model is absent.

- ε-Greedy

$$a^* \leftarrow \arg\max_a Q(s,a)$$

$$\forall a \in A(s) \quad \pi(a|s) \leftarrow \begin{cases} 1-\varepsilon+\varepsilon/|A(s)| & a=a^* \\ \varepsilon/|A(s)| & a \neq a^* \end{cases}$$

↳ any greedy policy wrt Q follows π is always an improvement over any ϵ-soft policy $\Rightarrow$ assurance from policy improvement theorem.

- SARSA $\rightarrow$ $\underbrace{\text{ON POLICY TD}}$

$\downarrow$

we try improving the current policy used to make decision, keep on improving

$\forall$ transi^n from non-terminal state,

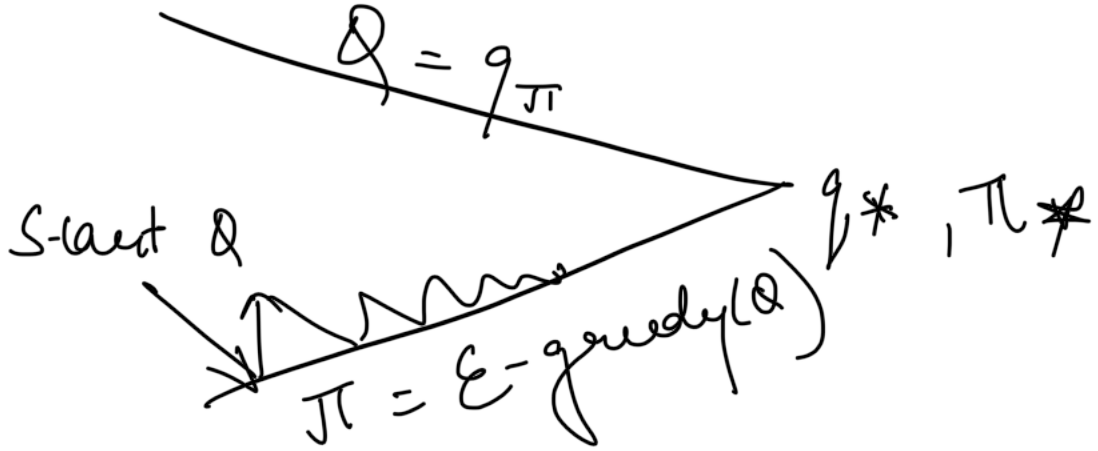
$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left[r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t) \right]$$

$\rightarrow$ TD(0) for Q

if s_{t+1} is T, $Q(s_{t+1}, a_{t+1}) = 0$

- How to guarantee convergence —
  1. all (s, a) pairs are visited ∞
  2. policy converges in the limit to the greedy policy.

remember to $s \leftarrow s'$, $a \leftarrow a'$



at each time step t,

1. Evaluation: SARSA, $Q \approx q_\pi$
2. Improvement: ε-greedy.

- sometimes, $s = s'$ $\Rightarrow$ leads to sampling from different distribution.

L4 Q-learning

trajectory is same

1 step Q learning ($\approx$ TD(0))

$$\hat{q}(s_t, a_t) \Leftarrow \hat{q}(s_t, a_t) + \alpha \overbrace{\left[r_{t+1} + \gamma \max_a \hat{q}(s_{t+1}, a) - \hat{q}(s_t, a) \right]}^{\Delta \text{ from TD}}$$

- stochastic averaging rule

$$Q^*(s,a) = E\left\{ r_{t+1} + \gamma \max_a Q^*(s_{t+1}, a') \,\middle|\, s_t = s,\ a_t = a \right\}$$

$$E(x) = \frac{1}{n}\sum_{i=1}^{n} x_i$$

$$\bar{x}_{n+1} = \bar{x}_n + \alpha(x_{n+1} - \bar{x}_n)$$

new estimate = old estimate + α (new sample − old estimate)

- Q learning → Off Policy → ε-greedy
  1. behavioral – generates behavior
  2. estimation – policy evaluates & improves.
     ↳ greedy

Eg. of cliff walking → SARSA vs Q learn.
- q learning ignores exploration. $\varepsilon = 0.1$
  It learns time optimal path.
- But SARSA, looks carefully and learns the safe path.
- q learns optimal policy which is greedy wrt to q-function.
- SARSA continues exploration.
- q learning helps in batch mode

learning.

L5 Off-Policy Learning

- we evaluate the target policy $\pi(a|s) \rightarrow$ calculate the $v_\pi(s)$ or $q_\pi(s,a)$
  - ↳ for this we have to follow behaviour policy $\mu(a|s)$
- how coverage is done?
  1. every ac^n taken under π is taken, sometimes, under μ.
  2. $\pi(a|s) > 0 \Rightarrow \mu(a|s) > 0$

  π can be greedy, μ can be ε-greedy.
- Importance Sampling $\rightarrow$ Technique to estimate the expected values under 1 distribu^n given it samples from each other.

$$E_{X \sim P}[f(X)] = \sum_x P(X) f(X)$$

$$= E_{X \sim Q}\left[\frac{P(X)}{Q(X)} f(X)\right]$$

- Off Policy MC

$$\rho_t^T = \prod_{k=1}^{T-1} \frac{\pi(A_k|S_k)}{\mu(A_k|S_k)}$$

$\pi(A_k|S_k) \rightarrow$ comp. prob. of trajectory

$\mu(A_k|S_k) \rightarrow$ probab. of getting same trajectory

Weighted average return for V.

$$V(s) = \frac{\sum_{t \in \mathcal{T}(s)} \rho_t^{T(t)} G_t}{\sum_{t \in \mathcal{T}(s)} \rho_t^{T(t)}}$$

with weighted average sampling