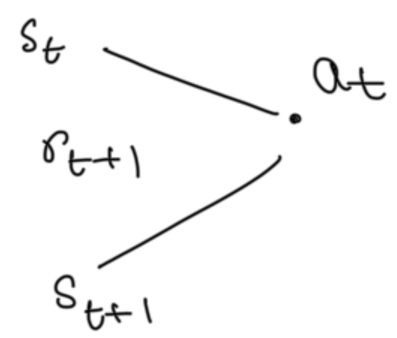
11 Apr 25

Simplest "TD" method



move a small step α in the direction of error

$$V(S_t) \leftarrow V(S_t) + \alpha\Big[\underbrace{r_{t+1} + \gamma V(S_{t+1})}_{\text{estimate of } V(S_t) \text{ but at time } t+1} - \underbrace{V(S_t)}_{\substack{V(S_t) \text{ at time } t \\ \text{my prediction as I land at } S_t}}\Big]$$

TD error δ

- TD like MC - do not require complete env
  only experience (sampling)
- can be fully incremental (bootstrapping)
- can learn even w/o the final outcome.

→ TD prediction

- $\pi \xrightarrow{get} v_\pi$ or q_π
- no knowledge of p and r but access to real system / sample model.

TD(0)

$$\underbrace{V_{k+1}(S_t)}_{\substack{\text{take } k \\ \text{steps of} \\ \text{updates}}} \leftarrow V_k(S_t) + \alpha\Big[\underbrace{r_{t+1} + \gamma V_k(S_{t+1})}_{\substack{\text{get sample} \\ \text{acc to} \\ \text{policy } \pi}} - V_k(S_t)\Big]$$

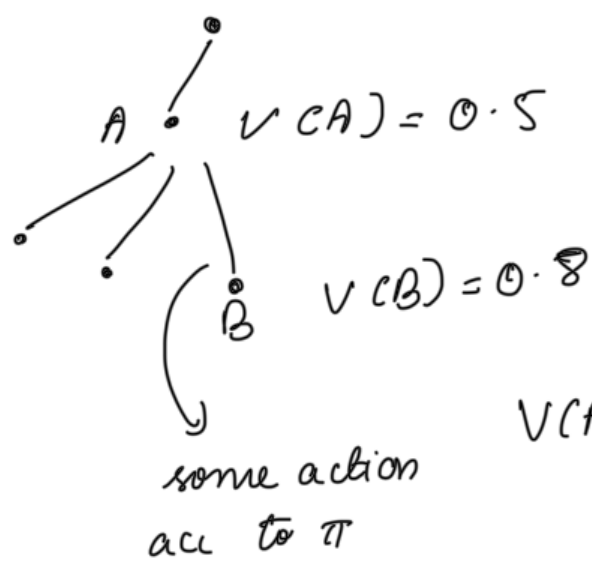
$V_k \rightarrow$ best estimate of V_π

$$V_\pi = E_\pi\{R_{t+1} + \gamma V_\pi(S_{t+1})\}$$

Transition acc to MDP, act acc to best policy.

MC | Exhaustive Search : Go all the way to the end.

TD(0) DP : Stop after a step (bootstrapping) eg. TD Update Example



r $A \to B : 0$

$\alpha : 0.2$ Step size

$\gamma = 0.9$ Discount factor

$$V(A) = V(A) + \alpha\left[R + \gamma V_B - V_A\right]$$

$$= 0.5 + 0.2\left[0.9(0.8) - 0.5\right]$$

$$= 0.544$$

→ MC vs TD Updates

Same trajectory, value function by MC vs TD

Batch TD: Take a batch / set of episodes
loop TD
till convergence

MC: first visit

fixed policy.

Consider: A · B

A0B0 (terminate)
B1
B0
A1B0
B1
B0

MC

$$V(A) = \frac{0+1}{2} = 0.5$$ look at A0B0, A1B0

$$V(B) = \frac{0+1+0+0+1+0}{6} = \frac{2}{6} = \frac{1}{3} = 0.33$$

$\gamma = 1$

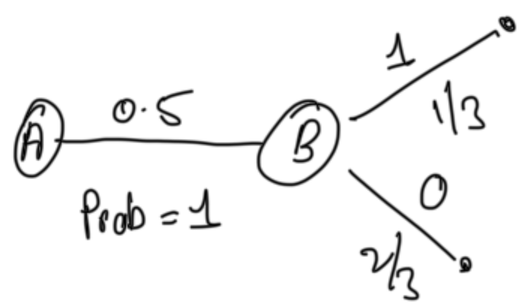
TD

$$V(A) = E\left[R + \gamma V(B)\right] = 1 + \gamma V(B) = 1 + \gamma\left(\frac{1}{3}\right)$$

on every time I go to B $= 0.833$

$$V(B) = E[R] = \frac{1}{3}$$

"implicitly" forming an MDP even though I am only given samples of an MDP.



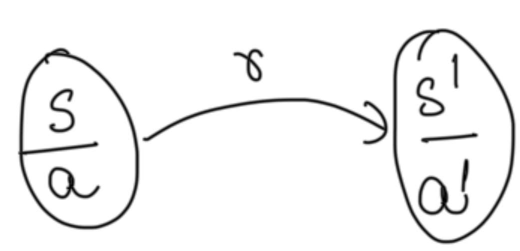
which is correct? happen because finite data.

Lot of data → both converge to same.

→ MC – converges to min least squares estimate of return.

TD – certainty equivalence estimate.

→ SARSA



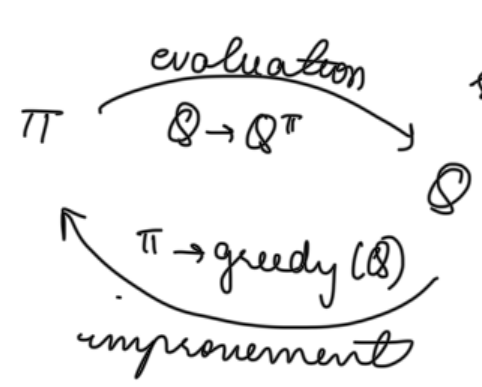
But Q learning we don't use next action

Look at TD evaluation

now TD control

find the optimal policy.

GPI:



some estimate

Policy Evaluation : use TD(0)

Policy Improvement : make greedy wrt current value function

Note:
We estimate action values rather than state values in the absence of model.

ε- Greedy Policies :

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

$\forall a \in \mathcal{A}(s)$:

$$\pi(a|s) = \begin{cases} 1-\varepsilon+\frac{\varepsilon}{|\mathcal{A}(s)|} & \text{if } a = a^* \\ \varepsilon / |\mathcal{A}(s)| & \text{if } a \neq a^* \end{cases}$$

→ any ε greedy policy wrt Q following π is an impr over any ε-soft policy is assured by the policy improvement theorem.

π(a|s) is atleast ε for every a

(1) SARSA : On-Policy TD control

samples acc to policy we are trying to evaluate & improve

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

TD(0) for Q

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

Sarsa Algorithm:

initialize Q(s, a) arbitrarily

For each episode

π : derived from Q

s

a = π(s) (eg. ε greedy)

$a = \pi(s)$

For each step in episode

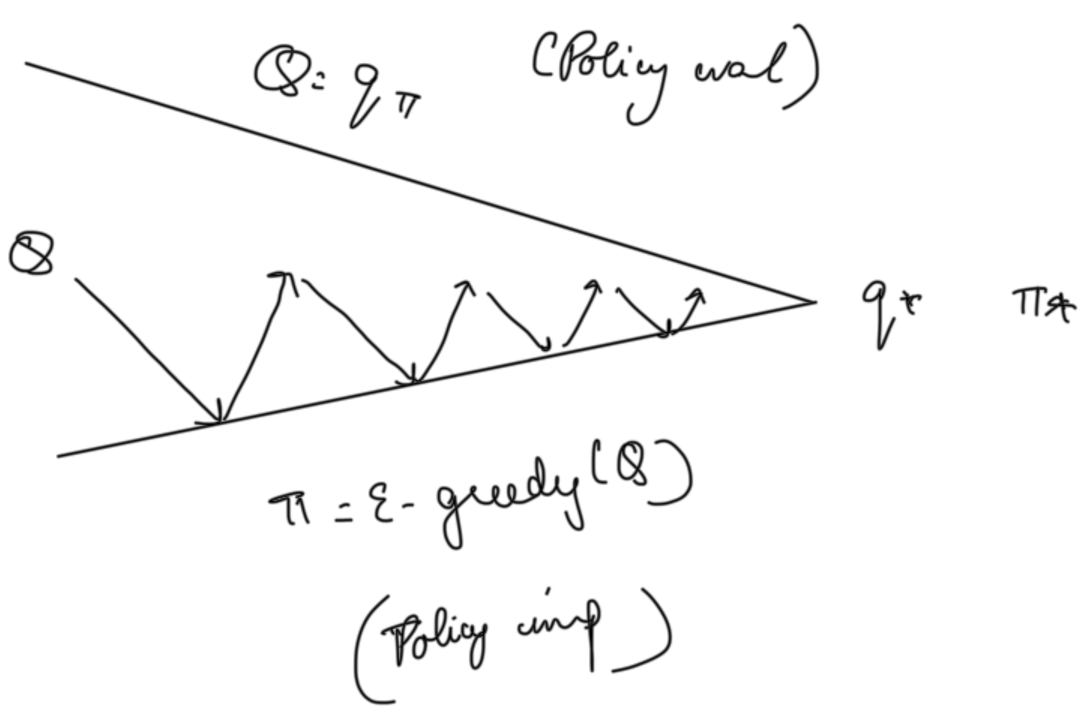
$$s \xrightarrow[r]{a} s'$$

(improvement) $\quad a' = \pi(s')$

(evaluation) $\quad Q(s,a) \leftarrow Q(s,a) + \alpha\left[r + \gamma Q(s',a') - Q(s,a)\right]$

$s \leftarrow s' \quad a \leftarrow a'$

until s is terminal



Convergence

- all (s,a) vis ∞
- converges in the limit to the greedy policy $\varepsilon \to 0$

(GLIE)

→ Q-learning

One-step Q-learning

$$\hat{q}(s_t, a_t) \leftarrow \hat{q}(s_t, a_t) + \alpha\Big[\underbrace{r_{t+1} + \gamma \underbrace{\max_a \hat{q}(s_{t+1}, a)} - \hat{q}(s_t, a_t)}_{}\Big]$$

Temporal Difference

in SARSA $Q(s_{t+1}, a_{t+1})$

Bellman Optimality Equation:

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

Stochastic Averaging Rule:

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

$$\overline{x_{n+1}} = \frac{1}{n+1}\left(x_{n+1} + \overline{x_n}\cdot n\right)$$

$$= \frac{1}{n+1}\left(x_{n+1} + (n+1)\overline{x_n} - \overline{x_n}\right)$$

$$= \overline{x_n} + \frac{1}{n+1}\left(x_{n+1} - \overline{x_n}\right)$$

$$= \overline{x_n} + \alpha\left(x_{n+1} - \overline{x_n}\right)$$

new est = old est + α (new sample - old est)

TD(0) → expectation in Bellman Eq to an average
Q learning → optimality

→ Q learning : Off policy TD control

Off policy

π

gen → Behavior
ϵ-greedy

eval, improve → estimation
greedy

Q - Learning Algorithm

initialize $Q(s,a)$ arbitrarily
for each episode
initialise s
for each step of ep

can be uniform random or anything you like

$a = \pi(s)$ (policy derived from Q eg ε-greedy)

$s \xrightarrow[r]{a} s'$

estimation policy is always greedy

$Q(s,a) \mathrel{+}= \alpha\left[r + \gamma \max_{a'} Q(s',a') - Q(s,a)\right]$

$s \leftarrow s'$

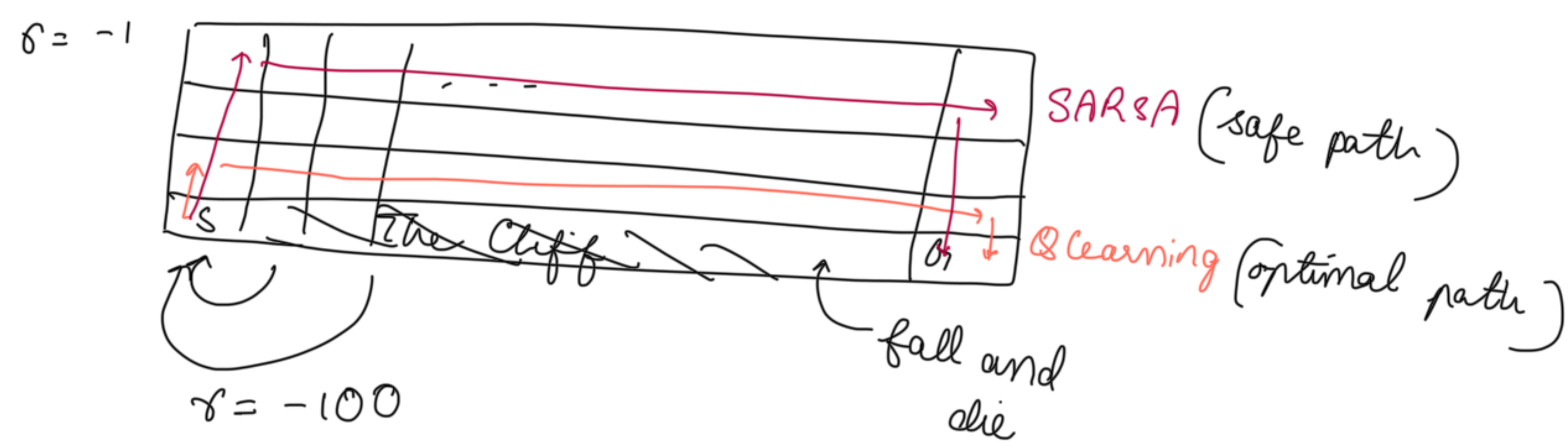
until s is terminal

→ if ε-greedy

only $\varepsilon - \frac{\varepsilon}{|A(s)|}$ times they differ

SARSA acc to exploratory action

Q greedy



say 10% times I decide to go to cliff, & fall and update my previous state in SARSA

Q-learning : still update acc to greedy policy.
assumes you will behave greedily in future

Sarsa : ε greedy execution in future.
(exploratory)
incorporate cost of exploration

Off Policy Learning

- Target $\pi(a|s) \xrightarrow{eval} V_\pi(s), q_\pi(s,a)$

policy

- Behaviour policy $\mu(a|s)$ eg. if π deterministic

observing somebody else (agent) but want to learn about π

- assumption of coverage:

  - $\pi(a|s) > 0 \rightarrow \mu(a|s) > 0$

- eg. π - greedy

  μ - ε-greedy

→ Importance Sampling:

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

$X \sim P$ but can only sample acc to Q

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

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

if I see it often in P but not Q it is importance to give it more weight and if $P(x) \neq 0$ imp $Q(x) \neq 0$

→ Importance Sampling Ratio

$$\rho_t^T = \frac{\prod_{k=t}^{T-1} \pi\left(\frac{A_k}{S_k}\right) p\left(\frac{S_{k+1}}{S_k A_k}\right)}{\prod_{k=t}^{T-1} \mu\left(\frac{A_k}{S_k}\right) p\left(\frac{S_{k+1}}{S_k, A_k}\right)}$$

$$= \frac{\prod_{k=t}^{T-1} \pi(A_k / S_k)}{\prod_{k=t}^{T-1} \mu(A_k / S_k)} = \log \sum \frac{\pi}{\mu}$$

$k=t$

weighted average return for V

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

Random Variable: $\underbrace{G(t)}_{\text{trying to sample \& average}}$

$\frac{N}{D}$

$$G(t) = r_{t+1} + \gamma r_{t+2} + \cdots + \gamma^{T-t-1}$$

$S_t \; A_t \; S_{t+1} \; \cdots$