Value function under policy π at state s.

$v_\pi(s) = \mathbb{E}_\pi[G_t \mid S_t = s]$ → solution to a set of equations.

$$= \sum_a \pi(a|s) \sum_{s',r} p(s', r | s, a) \left[r + v_\pi(s') \right] : s \in S$$

$v_*(s) = \max_\pi v_\pi(s)$ // optimal value function
↳ under optimal policy.

$$v_*(s) = \max_a \sum_{s',r} p(s', r | s, a) \left[r + v_* (s') \right]$$ → given & known optimal policy from s' state.

$$q_*(s, a) = \sum_{s',r} p(s', r | s, a) \left[r + \gamma \max_a q_*(s', a) \right]$$

* action-value function: value after taking action a. ← not/or be in policy.

Policy Improvement:

$$v_\pi(s, a) = \sum_{s',r} p(s', r | s, a) \left[r + v_\pi(s') \right]$$

if $s \in S$, $q_\pi(s, \pi'(s)) \geq v_\pi(s)$; [$\pi(s)$ and $\pi'(s)$ are two different policy]

then, $v_{\pi'}(s) \geq v_\pi(s)$

Greedy policy: $\pi'(s) = \arg\max_a q_\pi(s, a)$

$$= \arg\max_a \sum_{s',r} p(s', r | s, a) \left[r + \gamma v_\pi(s') \right]$$

$$v_{\pi'}(s) = \max_a \sum_{s',r} p(s', r | s, a) \left[r + \gamma v_\pi(s') \right]$$

(Improvement)

Policy iteration:

$\pi_0 \xrightarrow{E} v_{\pi_0} \xrightarrow{I} \pi_1 \xrightarrow{E} v_{\pi_1} \xrightarrow{I} \pi_2 \cdots \rightarrow \pi_* \rightarrow v_*$

(Evaluation)

↑ for all state

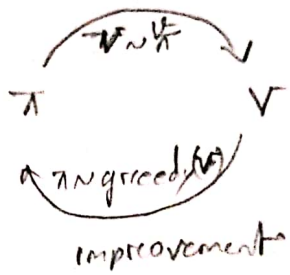
Value iteration:

Loop ⇒ $v_{k+1}(s) = \max_a \sum_{s',r} p(s', r \mid s, a)\left[r + \gamma v_k(s')\right]$ // greed action // sweep over (s)

after loop → $\pi(s) = \arg\max_a \sum_{s',r} p(s', r \mid s, a)\left[r + \gamma v_k(s')\right]$

$\left[\{v_k\}$ seq converge to $v_*(s)\right]$

Generalized Policy Evaluation:



Monte Carlo **prediction** :- estimate $V \approx v_\pi$ [episodic]

; DP alternative

policy π

$V(s) \in \mathbb{R}$

Returns(s) ← empty list ∀ s ∈ S.

loop forever; (for each episode):

generate $S_0, A_0, R_1, S_1, A_1, R_2 \cdots S_{T-1}, A_{T-1}, R_T$

$G \leftarrow 0$

t in range $(T-1, T-2, \ldots 0)$

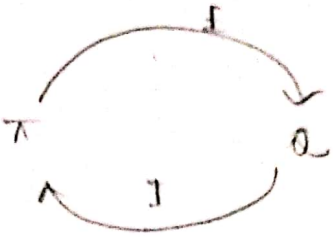
$G \leftarrow G + \gamma R_{t+1}$

unless S_t appear in $S_0, \ldots S_{t-1}$: (last S_t before reach to S_0 / 1st S_t appearance)

Append G to Returns S_t

$V(S_t) \leftarrow$ avg (Returns (S_t))

monte Carlo Control



$\pi_0 \xrightarrow{I} q_{\pi_0} \xrightarrow{I} \pi_1 \xrightarrow{E} q_{\pi_1} \xrightarrow{I} q_{\pi_0} \cdots \rightarrow \pi_* \rightarrow q_*$

$\pi(s) \doteq \arg\max_a q(s,a)$

$q_{\pi_k}(s, \pi_{k+1}(s)) = \max_a q_{\pi_k}(s,a)$

(policies π_k, π_{k+1} may vary in only one state)

MC (Exploring starts) $\pi \approx \pi_*$ (control) (also possible ϵ-greedy + Heuristic for s)

$\pi(s) \in A(s)$ (random) $s \in S$

$Q(s,a) \in \mathbb{R}$ (randomly) $a \in A(s)$

Returns $(s,a) \leftarrow$ empty list $s \in S$, $a \in A(s)$

Loop forever: (episodic)

choose $S_0 \in S$, $A_0 \in A(S_0)$; with prob > 0

generate episode from S_0, A_0 following $\pi: S_0, A_0, R_1 \cdots S_{T-1}, A_{T-1}, R_T$

$G \leftarrow 0$

Loop for $t = T-1, T-2, \cdots 0$

$G \leftarrow G + \gamma R_{t+1}$

Unless (S_t, A_t) pair appears earlier $(S_0, A_0 \cdots S_{t-1}, A_{t-1})$:

Append G to Returns (S_t, A_t)

$Q(S_t, A_t) = \text{avg}(\text{Returns}(S_t, A_t))$

$\pi(S_t) \leftarrow \arg\max_a Q(S_t, a)$ // deterministic

mc update $V(S_t) \leftarrow V(S_t) + \alpha(G_t - V(S_t))$

Temporal difference: $v(S_t) \leftarrow v(S_t) + \alpha\left[R_{t+1} + \gamma V(S_{t+1}) - v(S_t)\right]$

TD(0) to estimate V_π (vs DP) (value based)

- π
- $\alpha \in [0,1]$
- initialize $v(s)$, $\forall s \in S^+$, $v(\text{terminal}) = 0$;

Loop: for each episode
  initialize S
  Loop
    $A \leftarrow$ action by π
    take action A, observe R, S'
    $v(S) \leftarrow v(S) + \alpha\left[r + \gamma v(S') - v(S)\right]$
    $S \leftarrow S'$
  until S is terminal

TD error: $\delta_t = \left[R_{t+1} + \gamma v(S_{t+1}) - v(S_t)\right]$

$G_t - v(S_t)$: $\sum_{k=1}^{T-1} \gamma^{k-1} \delta_k$ //

Sarsa: TD on policy

$$Q(S_t, a_t) \leftarrow Q(S_t, a_t) + \alpha\left[R_{t+1} + \gamma Q(S_{t+1}, a_{t+1}) - Q(S_t, a_t)\right]$$

$\alpha \in [0,1]$

$Q(s,a)$ initialize, $s \in S^+$, $a \in A(s)$; $Q(\text{terminal}, \cdot) = 0$

Loop
  init S
  choose A from S using ε-greedy derive from Q
  loop for each episode
    Take A, observe S, R
    A' choose from S', ε-greedy derive from Q
    $Q(S,A) \leftarrow Q(S,A) + \alpha\left[R + \gamma Q(S', A') - Q(S, A)\right]$
    $S \leftarrow S'$, $A \leftarrow A'$
  until S is terminal

Q learning - off policy TD: $Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha\left[R_{t+1} + \gamma \max_{a \in A} Q(S_{t+1}, a) - Q(S_t, A_t)\right]$

$\alpha \in [0, 1]$

$Q(s, a)\ \forall\ s \in S,\ a \in A(s), S(\text{terminal}, \cdot) = 0$

loop for each
  init S
  Loop episode
    choose A from S using policy derived from Q (ε-greedy)
    Take action A observe reward R, S'

$$Q(S, A) \leftarrow Q(S, A) + \alpha\left[R_{t+1} + \gamma \max_a Q(S'_{t+1}, a) - Q(S_t, A_t)\right]$$

    $S \leftarrow S'$
  until S is terminal.

Expected SARSA:

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha\left[R_{t+1} + \gamma\, \mathbb{E}_\pi\left[Q(S_{t+1}, A_{t+1}) \mid S_{t+1}\right] - Q(S_t, A_t)\right]$$

$$\leftarrow Q(S_t, A_t) + \alpha\left[R_{t+1} + \gamma \sum_a \pi(a|S_{t+1})\, Q(S_{t+1}, a) - Q(S_t, A_t)\right]$$

Double Q learning:

$$Q_1(S_t, A_t) \leftarrow Q_1(S_t, A_t) + \alpha\left[R_{t+1} + \gamma Q_2\left(S_{t+1}, \arg\max_a Q_1(S_{t+1}, a)\right) - Q_1(S_t, A_t)\right]$$

Similar to Q learning → just change the update equation

(Q_1 is Q_2 with each prob of 0.5)

$G_t = R_{t+1} + \gamma V_t(S_{t+1}) + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots$

$G_{t:t+1} = R_{t+1} + \gamma V_t(S_{t+1})$

$G_{t:t+2} = R_{t+1} + \gamma R_{t+2} + \gamma^2 V_{t+1}(S_{t+2})$ → $t+1$ time policy

↓ final state, time

n step returns:

$$V_{t+n}(S_t) = V_{t+n-1}(S_t) + \alpha\left[G_{t:t+n} - V_{t+n-1}(S_t)\right]$$

n-step TD : $v \approx v_\pi$

Input: π

$\alpha \in [0,1]$, n

$V(s)$ arbitrary initialization.

All store and operation (for S_t and R_t) and their mod $(n+1)$

Loop: for each episode

init & store $S_0 \neq$ terminal

$T \leftarrow \infty$

Loop: $t = 0, 1, 2 \ldots$:

if $t < T$ then

action $\pi(\cdot|S_t)$

observe R_{t+1}, S_{t+1}

$S_{t+1} =$ terminal $\Rightarrow T \leftarrow t+1$

$\tau \leftarrow t-n+1$ // to be update.

if $\tau \geq 0$

$G \leftarrow \sum_{i=\tau+1}^{\min(\tau+n, T)} \gamma^{i-\tau-1} R_i$

if $\tau+n < T$ then $G \leftarrow G + \gamma^n V(S_{\tau+n})$

$V(S_\tau) \leftarrow V(S_\tau) + \alpha\left[G - V(S_\tau)\right]$

until $\tau = T-1$

Error reduction property of n-step return

$$\max_s \left| E_\pi\left[G_{t:t+n} \middle| S_t = s\right] - v_\pi(s) \right| \leq \gamma^n \max_s \left| V_{t+n-1}(s) - v_\pi(s) \right|$$

n step off policy learning: (importance sampling)

$$V_{t+n}(S_t) = V_{t+n-1}(S_t) + \alpha \rho_{t:t+n-1}\left[G_{t:t+n} - V_{t+n-1}(S_t)\right]$$

$$\rho_{t:h} = \prod_{k=t}^{\min(h, T-1)} \frac{\pi(A_k|S_k)}{b(A_k|S_k)} \quad \text{// importance ratio.}$$

init

loop: Algorithm: n step

init Q, $b(a|s)$

init π with n.t. greedy in Q

$\alpha \in (0, 1]$, n positive int.

store and access (S_t, A_t, R_t)

Loop:

init $S_0 \neq$ terminal

action $A_t \in b(\cdot|S_t)$

$T \leftarrow \infty$

Loop: $t = 0, 1, 2 \ldots$:

if $t < T$ then

A_t

observe R_{t+1}, S_{t+1}

if S_{t+1} terminal

$T \leftarrow t+1$

else: $A_{t+1} \sim b(\cdot|S_{t+1})$

$\tau \leftarrow t - n + 1$

if $\tau \geq 0$: $\rho \leftarrow \prod_{i=\tau+1}^{\min(\tau+n-1, T-1)} \frac{\pi(A_i|S_i)}{b(A_i|S_i)}$

$G \leftarrow \sum_{i=\tau+1}^{\min(\tau+n, T)} \gamma^{i-\tau-1} R_i$

if $\tau + n < T$ then $G \leftarrow G + \gamma^n Q(S_{\tau+n}, A_{\tau+n})$

$Q(S_\tau, A_\tau) \leftarrow Q(S_\tau, A_\tau) + \alpha \rho [G - Q(S_\tau, A_\tau)]$ ...

n-step SARSA

$$G_{t:t+n} \doteq R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^{n-1} R_{t+n} + \gamma^n Q_{t+n-1}(S_{t+n}, A_{t+n})$$

$$G_{t:t+n} = G_t \quad ; \; t+n \geq T$$

$$; \; n \geq 1; \; 0 \leq t \leq T-n$$

$$Q_{t+n}(S_t, A_t) \doteq Q_{t+n-1}(S_t, A_t) + \alpha \left[G_{t:t+n} - Q_{t+n-1}(S_t, A_t) \right]$$

n-step SARSA: $Q \approx q_*$ or q_π : n-step on policy

Init $Q(s,a)$; $s \in S$, $a \in A$: Arbitrarily

π init = ε greedy.

Param: $\alpha \in [0,1]$; small $\varepsilon > 0$; n

store and access (for S_t, A_t, R_t) to take $(n+1)$ mod

Loop:

ini $S_0 \neq$ terminal

select & store $A_0 \sim \pi(\cdot | S_0)$

$T \leftarrow \infty$

Loop for $t = 0, 1, 2 \ldots$

if $t < T$ then

Action A_t

observe R_{t+1}, S_{t+1}

S_{t+1} = terminal $\Rightarrow T = t+1$ and state.

else: $A_{t+1} \sim \pi(\cdot | S_{t+1})$

$\tau \leftarrow t+n-1$; (most updated time)

if $\tau \geq 0$: $\min(\tau+n, T)$, $i = \tau+1$

$G \leftarrow \sum_{i=\tau+1}^{\min(\tau+n, T)} \gamma^{i-\tau-1} R_i$

if $\tau+n < T$ then $G \leftarrow G + \gamma^n Q_{\tau+n}(S_{\tau+n}, A_{\tau+n})$

$Q(S_\tau, A_\tau) \leftarrow Q(S_\tau, A_\tau) + \alpha \left[G - Q(S_\tau, A_\tau) \right]$

if π is being learned, $\pi(\cdot | S_\tau)$ is ε greed w.r.t Q.

until $\tau = T-1$

Q-Learning & Expected Sarsa → 1 step algo. } off policy.

Tree backup → multistep Alg.

1step expected SARSA: $G_{t:t+1} = R_{t+1} + \gamma \sum_a \pi(a|S_{t+1}) Q_{t}(S_{t+1}, a)$

2 step: $G_{t:t+2} = R_{t+1} + \gamma \sum_{a \neq A_{t+1}} \pi(a|S_{t+1}) Q_{t+1}(S_{t+1}, a) + \gamma \pi(A_{t+1}|S_{t+1}) \underbrace{\left[R_{t+2} + \gamma \sum_a \pi(a|S_{t+2}) Q_{t+1}(S_{t+2}, a)\right]}_{G_{t+1:t+2}}$

n-step: $G_{t:t+n} = R_{t+1} + \gamma \sum_{a \neq A_{t+1}} \pi(a|S_{t+1}) Q_{t+n-1}(S_{t+1}, a) + \gamma \pi(A_{t+1}|S_{t+1}) G_{t+1:t+n}$

update value: $Q_{t+n}(S_t, a_t) = Q_{t+n-1}(S_t, A_t) + \alpha \left[G_{t:t+n} - Q_{t+n-1}(S_t, A_t)\right]$

Tree back Alg: $Q \approx q_*$, or q_π

init $Q(s,a)$ $s \in S$, $a \in A$, π, $\alpha \in (0,1]$, n, $\text{mod}(n+1)$

Loop: episode.
  init store S_0 & terminal
  select A_0, arbitrary S_0, store A_0
  $T \leftarrow \infty$
  Loop $t = 0, 1, 2 \ldots$
    if $t < T$:
      Action A_t obs R_{t+1}, S_{t+1}
      if S_{t+1} == terminal $T \leftarrow t+1$
      else: arbitrary A_{t+1} as S_{t+1}; store A_{t+1}
    $\tau \leftarrow t+1-n$
    if $\tau \geq 0$:
      if $t+1 \geq T$: $G \leftarrow R_T$
      else: $G \leftarrow R_{t+1} + \gamma \sum_a \pi(a|S_{t+1}) Q(S_{t+1}, a)$
      Loop: $k = \min(t, T-1)$ down through $\tau+1$
        $G \leftarrow R_k + \gamma \sum_{a \neq A_k} \pi(a|S_k) Q(S_{k+1}) + \gamma \pi(A_k|S_k) G$
      $Q(S_\tau, A_\tau) \leftarrow Q(S_\tau, A_\tau) + \alpha [G - Q(S_\tau, A_\tau)]$ ...

Planning & Learning method: model based → DP, heuristic search (Planning)
model free → MC, TD. (learning)

Model —Planning→ Policy.

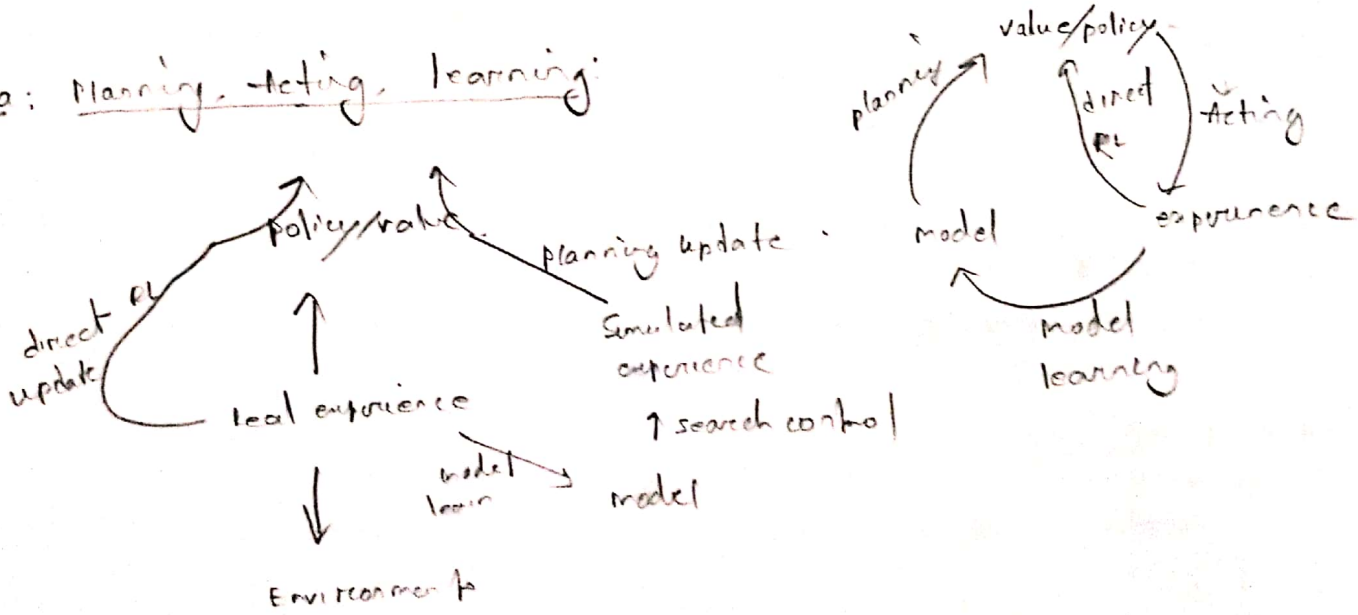
model → simulated Experience —backup→ value → policy.

Random-Sample one-step tabular Q-planning:

Loop forever:
1. select state $s \in S$ and $A \in A(s)$ at random
2. Send: s, A to sample model & obtain
   sample R, sample s'
3. Apply one step tabular Q learning.

$$Q(s,A) \leftarrow Q(s,A) + \alpha\left[R + \gamma \max_a Q(s',a) - Q(s,A)\right]$$

Dyna: Planning, Acting, learning



Tabular Dyna Q

Init: $Q(s,a)$, $model(s,a)$, $s \in S$, $a \in A(s)$

Loop forever:
a) $S \leftarrow$ current (nonterminal) state
b) $A \leftarrow$ ε-greedy(S,Q)
c) Take action A → observe R, S'
d) $Q(S,A) \leftarrow Q(S,A) + \alpha\left[R + \gamma \max_a Q(S',a) - Q(S,A)\right]$
e) $model(S,A) \leftarrow R, S'$ (Deterministic env)
f) Loop n times → $S \leftarrow$ random state, and $A \leftarrow$ action taken $R, S \leftarrow model(S,A)$
$Q(S,A) \leftarrow Q(S,A) + \alpha\left[R + \gamma \max_a Q(S',a) - Q(S,A)\right]$

Prioritized Sweeping :- Dyna Q

init $Q(s,a)$, model (s,a), $\forall$ s, a, PQueue to empty.

Loop forever:

a) $s \leftarrow$ nonterminal state

b) $A \leftarrow$ policy (S, Q)

c) Take A & observe R, S'

d) model $(s,a) \leftarrow R, S'$

e) $P \leftarrow \left| R + \gamma \max_a Q(S', a) - Q(S, A) \right|$

f) if $P > \theta$ then insert S, A into PQueue with priority P.

g) loop repeat n times while PQueue not empty:

$S, A \leftarrow$ first (PQueue)

$R, S' \leftarrow$ model (S, A)

$Q(S, A) \leftarrow Q(S, A) + \alpha \left[R + \gamma \max_a Q(S', a) - Q(S, A) \right]$

loop for $\forall$ $\bar{S}, \bar{A}$ predicted to lead to S:

$\bar{R} \leftarrow$ predicted reward $\bar{S}, \bar{A}, S$

$P \leftarrow \left| \bar{R} + \gamma \max_a Q(S, a) - Q(\bar{S}, \bar{A}) \right|$

if $P > \theta$ then insert $\bar{S}, \bar{A}$ into PQueue with priority

Approximate Solution method:

Parametric estimation: Approximate v_π given π

$\hat{v}(s, w) \approx v_\pi(s)$ // on memory.

under state $s \to u$ target update.

MC: $S_t \to G_t$; TD(0): $S_t \to R_{t+1} + \gamma \hat{v}(S_{t+1}, w)$

; TD(n): $S_t \to G_{t:t+n}$

DP policy evaluation $s \to E_\pi\left[R_{t+1} + \gamma \hat{v}(S_{t+1}, w) \mid S_t = s\right]$

Prediction Objective: mean square value Error: $\overline{VE}(w) = \sum_{s \in S} \mu(s)\left[v_\pi(s) - \hat{v}(s, w)\right]$

($\mu(s)$: time spent in s (if an policy, Distribution under π))

$\mu(s)$ estimation: $\eta(s) = h(s) + \sum_{\bar{s}} \eta(\bar{s}) \sum_a \pi(a|\bar{s})\, p(s|\bar{s}, a)$; $\bar{s} \in S$

$$\mu(s) = \frac{\eta(s)}{\sum_{\bar{s} \in S} \eta(\bar{s})} ;$$

Goal $\overline{VE}(w^*) \le \overline{VE}(w)$; $\forall w$

stochastic gradient & semi gradient:

$$w_{t+1} = w_t + \tfrac{1}{2}\alpha \nabla \left[v_\pi(S_t) - \hat{v}(S_t, w)\right]^2$$

$$= w_t + \tfrac{1}{2}\alpha \left[v_\pi(S_t) - \hat{v}(S_t, w)\right] \nabla_w \hat{v}(S_t, w)$$

$$\Rightarrow w_{t+1} = w_t + \alpha \left[U_t - \hat{v}(S_t, w)\right] \nabla_w \hat{v}(S_t, w)$$

$v_\pi(S_t) \doteq E\left[U_t \mid S_t = s\right]$ // unbiased case

Gradient monte Carlo algo. $\hat{v} \approx v_\pi$

Input: π policy

Input: Differentiable $\hat{v}: S \times \mathbb{R}^d \to \mathbb{R}$

Param $\alpha > 0$

Init $\underline{w} \in \mathbb{R}^d$

Loop forever:

Generate episode $S_0, A_0, R_1, S_1, A_1 \cdots R_T, S_T$ using π

Loop for each step of episode, $t = 0, 1, \cdots T-1$

$$\underline{w} \leftarrow \underline{w} + \alpha \left[G_t - \hat{v}(S_t, \underline{w}) \right] \nabla \hat{v}(S_t, \underline{w})$$

Semigradient: target: $U_t = R_{t+1} + \gamma \hat{v}(S_{t+1}, \underline{w})$

TD(0) semigradient algo:

Loop for each episode:

init S

Loop for each step of episode:

choose $A \sim \pi(\cdot|S)$

Take action A, observe R, S'

$$\underline{w} \leftarrow \underline{w} + \alpha \left[R + \gamma \hat{v}(S', \underline{w}) - \hat{v}(S, \underline{w}) \right] \nabla \hat{v}(S, \underline{w})$$

$S \leftarrow S'$

until S is terminal.

Linear methods

$$\hat{v}(s, \underline{w}) = \underline{w}^T \underline{x}(s) = \sum_{i=1}^{d} w_i x_i(s) \quad ; \forall s \quad \underline{x}(s) = (x_1(s), x_2(s) \cdots x_d(s))$$

$$\nabla \hat{v}(s, \underline{w}) = \underline{x}(s)$$

$$\Rightarrow \underline{w}_{t+1} = \underline{w}_t + \alpha \left[U_t - \hat{v}(S_t, \underline{w}_t) \right] \underline{x}(S_t)$$

Semi gradient TD(0) method: Linear.

$$w_{t+1} = w_t + \alpha\left[R_{t+1} + \gamma w_t^T x_{t+1} - w_t^T x_t\right] x_t$$

$$= w_t + \alpha\left[R_{t+1} x_t - x_t\left(x_t - \gamma x_{t+1}\right) w_t\right]$$

$$E\left[w_{t+1} \mid w_t\right] = w_t + \alpha\left[b - A w_t\right] ;$$

$$b = E\left[R_{t+1} x_t\right] \in \mathbb{R}^d$$

$$A = E\left[x_t\left(x_t - \gamma x_{t+1}\right)^T\right] \in \mathbb{R}^d \times \mathbb{R}^d$$

n-step: semi gradient TD

input: π

$\hat{v} : S^+ \times \mathbb{R}^d \to \mathbb{R}$: $\hat{v}(\text{terminal}) = 0$

Algo. param: $\alpha > 0$

$w \leftarrow$ init

All store and access (S_t & R_t) can

Loop for each episode:

init S_0 ≠ terminal

$T \leftarrow \infty$

Loop for $t = 0, 1, 2, \ldots$:

if $t < T$, then:

Take an action $\pi(\cdot \mid S_t)$

observe R_{t+1} & next state S_{t+1}

if S_{t+1} is terminal $T \leftarrow t+1$

$\tau \leftarrow t - n + 1$

if $\tau \geq 0$: $\min(\tau+n, T)$

$G \leftarrow \sum_{i=\tau+1}^{\quad} \gamma^{i-\tau-1} R_i$ $\quad \nabla \hat{v}(S_\tau, w)$

if $\tau + n < T$ then: $G \leftarrow G + \gamma^n \hat{v}(S_{\tau+n}, w)$ $w \leftarrow w + \alpha\left[G - \hat{v}(S_\tau, w)\right]$

$\hat{q}(s,a,\underline{w}) \doteq \underline{w}^T x(s,a) \doteq \sum_{i=1}^{d} w_i x_i(s,a)$

Semi gradient n-step Sarsa:

$$G_{t:t+n} \doteq R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^{n} R_{t+n} + \gamma^{n} \hat{q}(S_{t+n}, A_{t+n}, \underline{w}_{t+n-1})$$

$$\underline{w}_{t+n} \doteq \underline{w}_{t+n-1} + \alpha \left[G_{t:t+n} - \hat{q}(S_t, A_t, \underline{w}_{t+n-1}) \right] \nabla \hat{q}(S_t, A_t, \underline{w}_{t+n-1})$$

Episodic: semi gradient n-step SARSA: $\hat{q} \approx q_*$

Input: $\hat{q} : S \times A \times \mathbb{R}^d \to \mathbb{R}$

input: a policy π

$\alpha > 0, \epsilon > 0$

init value-function $\underline{w} \in \mathbb{R}^d$ arbitrarily (e.g., $\underline{w} = 0$)

store and access (S_t, A_t & R_t) take mode $(n+1)$

Loop: for each episode:

init & store $S_0 \neq$ terminal

select & store $A_0 \sim \pi(\cdot | S_0)$ or ϵ-greedy w.r.t. $\hat{q}(S_0, \cdot, \underline{w})$

$T \leftarrow \infty$

Loop for $t = 0, 1, 2, \ldots$

if $t < T$ then:

Take action A_t

observe & store next reward R_{t+1}, state S_{t+1}

if S_{t+1} is terminal

$T \leftarrow t+1$

else

select & store $A_{t+1} \sim \pi(\cdot | S_{t+1})$ or ϵ-greedy

$\tau \leftarrow t - n + 1$

if $\tau \geq 0$:

$G \leftarrow \sum_{i=\tau+1}^{\min(\tau+n, T)} \gamma^{i-\tau-1} R_i$

if $\tau + n < T$ then $G \leftarrow G + \gamma^{n} \hat{q}(S_{\tau+n}, A_{\tau+n}, \underline{w})$

$\underline{w} \leftarrow \underline{w} + \alpha \left[G - \hat{q}(S_\tau, A_\tau, \underline{w}) \right] \nabla \hat{q}(S_\tau, A_\tau, \underline{w})$

until $\tau = T - 1$

$$w_{t+n} \doteq w_{t+n-1} + \alpha \left[G_{t:t+n} - \hat{v}(s_t, w_{t+n-1}) \right] \nabla \hat{v}(s_t, w_{t+n-1})$$

$$G_{t:t+n} = R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^{n-1} R_{t+n} + \gamma^n \hat{v}(s_{t+n}, w_{t+n-1}), \quad 0 \le t < T$$

On policy Control with Approximation :-

$$q_*(s, a, w) = q_*(s, a) \; ; \; w \in \mathbb{R}^d$$

Episodic Semi-gradient Control

$$w_{t+1} \doteq w_t + \alpha \left[U_t - \hat{q}(s_t, A_t, w_t) \right] \nabla \hat{q}(s_t, A_t, w)$$

One step SARSA :

$$w_{t+1} \doteq w_t + \alpha \left[R_{t+1} + \gamma \hat{q}(s_{t+1}, A_{t+1}, w_t) - \hat{q}(s_t, A_t, w) \right] \nabla \hat{q}(s_t, A_t, w)$$

Semi gradient SARSA : $\hat{q} \approx q^*$

Input : action value $\hat{q} : S \times A \times \mathbb{R}^d \to \mathbb{R}$

$\alpha > 0, \epsilon > 0$

$w \in \mathbb{R}^d$ arbitrary.

Loop for each episode :

$S, A \leftarrow$ init state & action of episode (ϵ-greedy)

Loop for each step:

Take action A observe R, S'

if S' is terminal

$w \leftarrow w + \alpha \left[R - \hat{q}(S, A, w) \right] \nabla \hat{q}(S, A, w)$

go next episode.

choose A' as function $\hat{q}(S', \cdot, w)$; ϵ-greedy

$w \leftarrow w + \alpha \left[R + \gamma \hat{q}(S', A', w) - \hat{q}(S, A, w) \right] \nabla \hat{q}(S, A, w)$

$S \leftarrow S'$

$A \leftarrow A'$

$$r(\pi) \doteq \lim_{h\to\infty} \frac{1}{h} \sum_{t=1}^{h} \mathbb{E}\left[R_t \mid S_0, A_{0:t-1} \sim \pi\right]$$

$$= \lim_{t\to\infty} \mathbb{E}\left[R_t \mid S_0, A_{0:t-1} \sim \pi\right]$$

$$= \sum_s \mu_\pi(s) \sum_a \pi(a|s) \sum_{s',r} p(s', r \mid s, a)\, r$$

$$\sum_s \mu_\pi(s) \sum_a \pi(a|s)\, p(s'|s,a) = \mu_\pi(s')$$

$$G_t \doteq R_{t+1} - r(\pi) + R_{t+2} - r(\pi) + \cdots + R_{t+n} - r(\pi) + \cdots$$

// differential return & value.

$$v_\pi(s) \doteq \mathbb{E}_\pi\left[G_t \mid S_t = s\right]$$

$$q_\pi(s,a) \doteq \mathbb{E}_\pi\left[G_t \mid S_t = s, A_t = a\right]$$

→ Some definitions.

$$v_\pi(s) = \sum_a \pi(a|s) \sum_{r,s'} p(s', r \mid s, a)\left[r - r(\pi) + v_\pi(s')\right]$$

$$q_\pi(s,a) = \sum_{r,s'} p(s', r \mid s, a)\left[r - r(\pi) + \sum_{a'} \pi(a'|s')\, q_\pi(s', a')\right]$$

$$v_*(s) = \max_a \sum_{r,s'} p(s', r \mid s, a)\left[r - \max_\pi r(\pi) + v_*^\pi(s')\right]$$

$$q_*(s,a) = \sum_{r,s'} p(s', r \mid s, a)\left[r - \max_\pi r(\pi) + \max_{a'} q_*(s', a')\right]$$

→ $\delta_{-1} \doteq R_{t+1} - \bar{R}_t + \hat{v}(S_{t+1}, \underline{w}_t) - \hat{v}(S_t, \underline{w}_t)$

$$\delta_t \doteq R_{t+1} - \bar{R}_t + \hat{q}(S_{t+1}, A_{t+1}, \underline{w}_t) - \hat{q}(S_t, A_t, \underline{w}_t)$$

update: $\underline{w}_{t+1} \doteq \underline{w}_t + \alpha \delta_t \nabla \hat{q}(S_t, A_t, \underline{w})$

Semi gradient SARSA: $q \approx q_*$

Input $\hat{q} : S \times A \times \mathbb{R}^d \to \mathbb{R}$

$\alpha, \beta > 0$

init $\underline{w} \in \mathbb{R}^d$, $[\underline{w} = 0]$

init $\bar{R} \in \mathbb{R}$

init state s, action A

Loop for each step:

Take action A, observe, R', s'

choose A' as function of $\hat{q}(s', \cdot, \underline{w})$ // ϵ greedy .

$\delta \leftarrow R - \bar{R} + \hat{q}(S', A', \underline{w}) - q(s, A, \underline{w})$

$\bar{R} \leftarrow \bar{R} + \beta \delta$

$\underline{w} \leftarrow \underline{w} + \alpha \delta \nabla \hat{q}(s, A, \underline{w})$

$S \leftarrow S'$

$A \leftarrow A'$

Differential Semigradient:

$$G_{t:t+n} \doteq R_{t+1} - \bar{R}_{t+n-1} + \cdots + R_{t+n} - \bar{R}_{t+n-1} + \hat{q}(S_{t+n}, A_{t+n}, \underline{w}_{t+n-1})$$

$t+n \geq T$ then $G_{t:t+n} = G_t$

$$\delta_t \doteq G_{t:t+n} - \hat{q}(S_t, A_t, \underline{w})$$

Differential Semi gradient algorithm:

store & access (S_t, A_t, R_t) in mod $(n+1)$

init S_0, A_0

Loop : $t = 0, 1, 2 \cdots$

take action A_t

observe & store next reward R_{t+1} & next stage S_{t+1}

select $A_t \sim \pi(\cdot | S_{t+1})$ // ϵ - greedy

$\tau \leftarrow t - n + 1$

if $\tau \geq 0$: $\delta \leftarrow \sum_{i=\tau+1}^{\tau+n} (R_i - \bar{R}) + \hat{q}(S_{\tau+n}, A_{\tau+n}, \underline{w}) - \hat{q}(S_\tau, A_\tau, \underline{w})$

$\bar{R} \leftarrow \bar{R} + \beta \delta$

$\underline{w} \leftarrow \underline{w} + \alpha \delta \nabla \hat{q}(S_\tau, A, \underline{w})$

Function Approximation: Linear / ANN

Bootstrapping: DP, TD, MC (update target & estimates)

Off-policy training: Other than target policy.

combination of these cause instability.

Eligibility trace: TD(λ) → Q & SARSA:

[λ → 0 ; → TD(0) method] : [λ → 1 ; MC method] ; balance between TD & MC.

Eligibility trace $\underline{z}_t$; short term vector parallels ; long term weight $\underline{w}_t$

Trace decay $\lambda \in [0,1]$.

λ return; $G_t^\lambda \doteq (1-\lambda) \sum_{n=1}^{\infty} \lambda^{n-1} G_{t:t+n}$

$$G_t^\lambda = (1-\lambda) \sum_{n=1}^{T-t-1} \lambda^{n-1} G_{t:t+n} + \lambda^{T-t-1} G_t$$

$$\underline{w}_{t+1} \doteq \underline{w}_t + \alpha \left[G_t^\lambda - \hat{v}(s, \underline{w}) \right] \nabla_{\underline{w}} \hat{v}(s, \underline{w})$$

Semi gradient TD (λ):

Input : λ

input : differentiable $\hat{v} : S^+ \times \mathbb{R}^d \to \mathbb{R}$ $\quad \hat{v}(\text{terminal}, \cdot) = 0$

$\alpha > 0$, $\lambda \in [0,1]$

$\underline{w} \in$ Arbitrary.

loop:
  init S
  $\underline{z} \leftarrow 0$
  loop:
    choose $A \sim \pi(\cdot|s)$
    Take A, observe R, S'
    $\underline{z} \leftarrow \gamma\lambda \underline{z} + \nabla \hat{v}(s, \underline{w})$
    $\delta \leftarrow R + \gamma \hat{v}(s', \underline{w}) - \hat{v}(s, \underline{w})$
    $\underline{w} \leftarrow \underline{w} + \alpha \delta \underline{z}$
    $s \leftarrow s'$
  until s' is terminal.

n-step truncated λ-return Methods: TTD(λ)

$$G^{\lambda}_{t:h} = (1-\lambda)\sum_{n=1}^{h-t-1}\lambda^{n-1}G_{t:t+n} + \lambda^{h-t-1}G_{t:h}$$

$$w_{t+n} = w_{t+n-1} + \alpha\left[G^{\lambda}_{t:t+n} - \hat{v}(s_t, w_{t+n-1})\right]\nabla\hat{v}(s_t, w_{t+n-1})$$

$$G^{\lambda}_{t:t+k} = \hat{v}(s_t, w_{t-1}) + \sum_{i=t}^{t+k-1}(\gamma\lambda)^{i-t}\delta'_i$$

where, $\delta'_t = R_{t+1} + \gamma\hat{v}(s_{t+1}, w_t) - \hat{v}(s_t, w_{t-1})$

online λ-return Algorithm:

$$w^h_{t+1} = w^h_t + \alpha\left[G^{\lambda}_{t:h} - \hat{v}(s_t, w_t)\right]\nabla\hat{v}(s_t, w_t) \quad ; 0 \leq t < h \leq T$$

True online TD(λ) $[w^T x = v_x]$

Input π

Input: $x: S^+ \to \mathbb{R}^d$

$\alpha > 0$, $\lambda \in [0,1]$

Loop:
init state and get feat vector x

$z \leftarrow 0$

$V_{old} \leftarrow 0$

Loop:

- chose $A \sim \pi$
- Take action A, observe R, x'
- $V \leftarrow w^T x$
- $V' \leftarrow w^T x'$
- $\delta \leftarrow R + \gamma V' - V$
- $z \leftarrow \gamma\lambda z + (1 - \alpha\gamma\lambda z^T x)\, x$
- $w \leftarrow w + \alpha(\delta + V - V_{old})z - \alpha(V - V_{old})x$
- $V_{old} \leftarrow V'$
- $x \leftarrow x'$

until $x' = 0$ (terminal)

SARSA(λ) $w^T x \approx q_\pi$ or q_*

Input: $f(s,a)$ // Return feats. for s & a.

Input: policy π

Algo param $\alpha > 0$, $\lambda \in [0,1]$

init $\underline{w} = (w_1 \dots w_d)^T \in \mathbb{R}^d$; $\underline{z} \in \mathbb{R}^d$

Loop: for episode.

init S

choose $A \sim \pi(\cdot|s)$ ϵ-greedy to $\hat{q}(s,\cdot,\underline{w})$

$\underline{z} \leftarrow 0$

Loop for each step:

Take action $A \to R, S'$

$\delta \leftarrow R$

Loop for i in $f(s,A)$:

$\delta \leftarrow \delta - w_i$

$z_i \leftarrow z_i + 1$ // Accumulating trace

or $z_i \leftarrow 1$ // Replacing trace

if S' is terminal

$\underline{w} \leftarrow \underline{w} + \alpha \delta \underline{z}$

Go to next episode

choose $A' \sim \pi(\cdot|S')$ or near greedily $\sim \hat{q}(S',\cdot,\underline{w})$

Loop for i in $f(S',A')$: $\delta \leftarrow \delta + \gamma w_i$

$\underline{w} \leftarrow \underline{w} + \alpha \delta \underline{z}$

$\underline{z} \leftarrow \gamma \lambda \underline{z}$

$S \leftarrow S'$; $A \leftarrow A'$

True online SARSA(λ) with π

Input: $x : S^+ \times A \to \mathbb{R}^d$ $\quad x(\text{terminal}, \cdot) = 0$

Input: π

$\alpha > 0$; α, $\lambda \in [0, 1]$

Init $\underline{w} \in \mathbb{R}^d$

Loop for each episode:

Init S

Choose $A \sim \pi(\cdot|S)$

$x \leftarrow x(S, A)$

$z \leftarrow 0$

$Q_{old} \leftarrow 0$

Loop for each step:

Take $A \to R, S'$

Choose $A' \sim \pi(\cdot|S')$ near greedily S' using $\underline{w}$

$x' \leftarrow x(S', A')$

$Q \leftarrow \underline{w}^T x$

$Q' \leftarrow \underline{w}^T x'$

$\delta \leftarrow R + \gamma Q' - Q$

$z \leftarrow \gamma \lambda z + (1 - \alpha \gamma \lambda z^T x) x$

$\underline{w} \leftarrow \underline{w} + \alpha(\delta + Q - Q_{old})\underline{z} - \alpha(Q - Q_{old}) x$

$Q_{old} \leftarrow Q'$

$x \leftarrow x'$

$A \leftarrow A'$

until S is terminal.

semi gradient method:

$$\rho_t \doteq \rho_{t:t} \doteq \frac{\pi(A_t|S_t)}{b(A_t|S_t)}$$

$$w_{t+1} \doteq w_t + \alpha \rho_t \delta_t \nabla \hat{v}(S_t, w_t)$$

$$\delta_t \doteq R_{t+1} + \gamma \hat{v}(S_{t+1}, w_t) - \hat{v}(S_t, w_t) \quad \text{// continuous}$$

$$\delta_t \doteq R_{t+1} - \bar{R}_t + \hat{v}(S_{t+1}, w_t) - \hat{v}(S_t, w_t) \quad \text{// episodic}$$

Expected SARSA:

$$w_{t+1} \doteq w_t + \alpha \delta_t \nabla \hat{q}(S_t, A_t, w_t)$$

$$\delta_t \doteq R_{t+1} + \gamma \sum_a \pi(a|S_{t+1}) \hat{q}(S_{t+1}, a, w_t) - \hat{q}(S_t, A_t, w_t)$$
// episodic

$$\delta_t \doteq R_{t+1} - \bar{R}_t + \sum_a \pi(a|S_{t+1}) \hat{q}(S_{t+1}, a, w_t) - \hat{q}(S_t, A_t, w_t)$$
// continuous.

n-step version of semi gradient expected SARSA:

$$w_{t+n} = w_{t+n-1} + \alpha \rho_{t+1} \cdots \rho_{t+n-1} \left[G_{t:t+n} - \hat{q}(S_t, A_t, w_{t+n-1}) \right] \nabla \hat{q}(S_t, A_t, w_{t+n-1})$$

$$G_{t:t+n} = R_{t+1} + \cdots + \gamma^{n-1} R_{t+n} + \gamma^n \hat{q}(S_{t+n}, A_{t+n}, w_{t+n-1}) \quad \text{// episodic}$$

$$G_{t:t+n} = R_{t+1} - \bar{R}_t + \cdots + R_{t+n} - \bar{R}_{t+n-1} + \hat{q}(S_{t+n}, A_{t+n}, w_{t+n-1}) \quad \text{// cont.}$$

n-step tree backup algo.

$$w_{t+n} = w_{t+n-1} + \alpha \left[G_{t:t+n} - \hat{q}(S_t, A_t, w_{t+n-1}) \right] \nabla \hat{q}(S_t, A_t, w_{t+n-1})$$

$$G_{t:t+n} = \hat{q}(S_t, A_t, w_{t-1}) + \sum_{k=t}^{t+n-1} \delta_k \prod_{i=t+1}^{k} \gamma \pi(A_i|S_i)$$