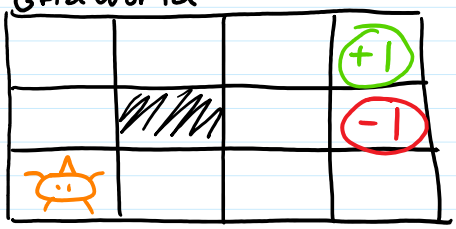
Temporal Difference (TD) Learning

GridWorld



at each step go ↑, ↓, ←, → .8 go intended dir .2 go to side

S = set of states

state (row, col)

next state

model

$P(s, s', a)$ = probability of action a in state s resulting in s' (action $A = \{\uparrow, \downarrow, \leftarrow, \rightarrow\}$)

$R(s, s', a)$ = reward from $s \xrightarrow{a} s'$

policy: function $\pi: S \rightarrow A$

$V_\pi(s)$ = expected future discounted reward given that in state s using π

π_{OPT}: policy that maximizes $V_{\pi_{OPT}}(s_0)$ (s_0 = initial state)

$V(s) = V_{\pi_{OPT}}(s)$

Value iteration (= introduce turn limit)

$Q(s,a)$

generally $V(s) = \max_a \sum_{s'} P(s,s',a) \cdot (R(s,s',a) + \gamma V(s'))$

($R(s,s',a)$: immediate reward; γ: discount; $V(s')$: future reward)

$\pi_{OPT}(s) = \text{argmax}_a Q(s,a)$

$V(r, c, n)$ = value of state (r, c) w/ n steps left

$$= \begin{cases} -1 & \text{if } r,c = (1,3) \\ +1 & \text{if } r,c = (0,3) \\ 0 & \text{if } n = 0 \\ \max_a & .8 \cdot V((r,c)+a, n-1) \\ & +.1\, V((r,c)+\overrightarrow{a}, n-1) \\ & +.1\, V((r,c)+\overleftarrow{a}, n-1) \end{cases}$$

$V(r,c,n)$ converges to $V(r,c)$ as $n \uparrow$

TD Value Learning

observe $s \xrightarrow{a = \pi(s)} s'$ giving immediate reward $R(s,s',a)$

get discounted future reward $\gamma \cdot V^\pi(s')$

sample of $V^\pi(s) = R(s,s',a) + \gamma V^\pi(s')$

update estimate $V^\pi(s) \leftarrow V^\pi(s) + \underbrace{\alpha}_{\text{learning rate}} \left(\underbrace{R(s,s',a) + \gamma V^\pi(s') - V^\pi(s)}_{\text{error}} \right)$

$= (1-\alpha) V^\pi(s) + \alpha \left[R(s,s',a) + \gamma V^\pi(s') \right]$

Q Learning

$Q(s,a)$ = estimate of expected discounted future reward

$V(s) \approx \max_a Q(s,a)$

initialize $Q(s,a) = \begin{cases} R(s) & \text{for terminal } s \\ 0 & \text{otherwise} \end{cases}$ (if you know this)

while not done

episode:

$s \leftarrow s_0$ (initial state)

while s not terminal

choose action a ← ε-greedy is good: prob ε choose a randomly uniformly; $1-\varepsilon$ choose $\operatorname{argmax}_a Q(s,a)$

observe transition (s, a, r, s') (r: immediate reward $(R(s,s',a))$)

update $Q(s,a) \leftarrow Q(s,a) + \alpha \left(r + \gamma \max_{a'} Q(s',a') - Q(s,a) \right)$

α: learning rate (can ↓ as episodes ↑)

$r + \gamma \max_{a'} Q(s',a') - Q(s,a)$: surprise (error)

Q-Yahtzee

unused categories, upper total, yahtzee bonus — rerolls, current roll

$\approx 2^{17} \cdot 64 \cdot \underbrace{3 \cdot 3}_{\approx 2^3} \cdot \underbrace{2^8}_{252 \approx 256} \approx 2^{29}$ ≈ 512 million

$Q(s,a)$ = estimate of expected value of action a in state s

6 ... 32, 1 ... 13 — reroll, choose cat — avg $\approx 2^4$

size of Q table

$\approx 2^{33}$ 8 billion

will take a long time

Generalization:

Ones	1
Twos	4
Threes	
Fours	
Fives	20
Sixes	18
3K	23
4K	0
FH	
SS	30
LS	
C	
Y	50

after roll 1

s_1 =

Ones	2
Twos	4
Threes	
Fours	16
Fives	
Sixes	18
3K	17
4K	0
FH	
SS	0
LS	
C	
Y	50

after roll 1

s_2 =

$Q(s_1, a) \approx Q(s_2, a)$

Function Approximators

- learn a fxn that approximates $Q(s,a)$

Linear Approximator

Define features of states and possibly actions

can ignore action for features of state only

$f_1(s,a)$ on pace to earn upper bonus

$f_2(s,a)$ is chance unused

⋮ both LS, SS unused

⋮ upper category a is unused

$$Q(s,a) = w_1 \cdot f_1(s,a) + \cdots + w_n f_n(s,a)$$

In state s

Choose action a using exploit/explore policy

Observe transition (s, a, r, s')

Update ~~$Q(s,a) \leftarrow Q(s,a) + \alpha(\max_{a'} Q(s',a') - Q(s,a))$~~

$$w_i \leftarrow w_i + \alpha\left(r + \gamma \max_{a'} Q(s',a') - Q(s,a)\right) \cdot f_i(s,a)$$

Given feature of states $f : S \to \mathbb{R}$

$f_a : S \times A \to \mathbb{R}$ $\quad f_a(s, a') = \begin{cases} f(s) & \text{if } a' = a \\ 0.0 & \text{otherwise} \end{cases}$

Beware: Q-learning may diverge
decrease α as time $\uparrow$
keep features on same scale as reward

Football: which position is better

(30, 4, 10, 6) — 1-10 from 30 w/ 30 sec remaining
or
(50, 4, 10, 20) — 1-10 from 50 w/ 100 sec remaining

(35, 3, 20, 12) — 2-20 from 35 w/ 60 sec
or
(35, 2, 5, 12) — 3-5 from 35 w/ 60 sec