n-step Bootstrapping

- Bridge TD(0) & MC
+ allow smooth transition based on task
- detach from timestep

+ Bootstrapping works best over large time periods / more change
+ TD(n) removes timestep boundary
- used for eligibility traces, but discussed later
- prediction problem first

n-step TD Prediction

- again MC = all rewards to evaluate
- TD(0) = 1 step
- n-step = TD of n steps

- Consider update for S_t from sequence $S_t, R_{t+1}, S_{t+1}, \dots R_T, S_T$

+ MC, needs complete return:

$$G_T \doteq R_{t+1} + \gamma R_{t+2} + \dots + \gamma^{T-t-1} R_T$$

+ MC, return is target of updates

+ TD(0) updates reward + discounted est value of next state one-step return

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

+ So, now let go to n=2, n=3

$$G_{t:t+2} \doteq R_{t+1} + \gamma R_{t+2} + \gamma^2 V_{t+1}(S_{t+2})$$

$$G_{t:t+n} \doteq R_{t+1} + \gamma R_{t+2} + \dots + \gamma^{n-1} R_{t+n} + \gamma^n V_{t+n-1}(S_{t+n})$$

- basically use discounted reward and "truncated" return at $V_{t+n-1}(S_{t+n})$

+ if $t+n \geq T$, $G_{t:t+n} \doteq G_t$ (terms beyond $T = 0$)

- can't really use though because R_{t+n} & V_{t+n-1} aren't known until seen <u>at</u> t+n

+ thus

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

and $V_{t+n}(s) = V_{t+n-1}(s) \; \forall \; s \neq S_t$

× can't make updates till n - 1 steps

× make additional n-1 updates after termination

EXERCISE 7.1

$$G_{t:t+n} - V(S_t) = R_{t+1} + \gamma G_{t+1:t+n} - V(S_t)$$

$$- \gamma V(S_{t+1}) + \gamma V(S_{t+1})$$

$$= \delta_t + \gamma(G_{t+1:t+n} - V(S_t))$$

$$= \delta_t + \gamma(\delta_{t+1} + \gamma(G_{t+2:t+n} - V(S_{t+2})))$$

$$= \delta_t + \gamma\delta_{t+1} + \gamma^2(G_{t+2:t+n} - V(S_{t+2}))$$

$$= \delta_t + \gamma\delta_{t+1} + \dots + \gamma^{n-1}(R_{t+n} + \gamma V(S_{t+n}) - V(S_{t+n-1}))$$

$$= \boxed{\sum_{k=t}^{t+n-1} \gamma^{k-t} \delta_k}$$

EXAMPLE 7.1

reasoning example for n-step TD

$n=2$ $T=3$

$t=0$ $\tau = t-n+1 = -1$ do nothing

$t=1$ $\tau=0$ $G = 1 \to 2$ $\gamma^0 R_1 + \gamma^1 R_2 + \gamma^2 V(S_2)$

$t=2$ $\tau=1$ $G = 2\text{-}3$ $\gamma^0 R_2 + \gamma^1 R_3$

$t=3$ $\tau=2$ $G = 3\text{-}3$ $\gamma^0 R_3$

- How do determine a good n value?
- Empirical analysis

+ examples here found intermediate values best

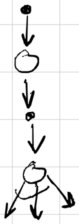
n-step SARSA

- combine it with control methods
- previous was one-step SARSA or SARSA(0)
- first redefine with action-values

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

- n-step expected SARSA:

+ single sequence till last state, then $V_{t+n-1}(S_{t+n})$



× $G_{t:t+n} \doteq R_{t+1} + \dots + \gamma^{n-1} R_{t+n} + \gamma^{n} \bar{V}_{t+n-1}(S_{t+n})$

with $\bar{V}_t(s)$ as expected approximate value

$$\bar{V}_t(s) \doteq \sum_a \pi(a|s)\, Q_t(s, a)$$

* used throughout later algos
* if s is terminal, $\bar{V}_t(s) = 0$

n-step Off Policy Learning

- n-step TD
- $$V_{t+n}(S_t) \doteq V_{t+n-1}(S_t) + \alpha \rho_{t:t+n-1} \left[G_{t:t+n} - V_{t+n-1}(S_t) \right] \quad 0 \le t \le T$$

+ $\rho_{t:t+n-1}$ because last step is only the state visited

x $V=0$ if ρ is 0 because behavior policy takes action π would never take

- for action values

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

- $\rho_{t+1:t+n}$ because A_t already taken, and

A_{t+n} has been chosen

- these generalize because if on-policy $\rho = 1$
- n-step expected SARSA uses $\rho_{t+n:t+n-1}$ because last value is V, across all $\underline{a}$

Per Decision Methods with Control Variates

- previous algos are not efficient with off policy sampling ratio

+ use notation $G_{t:h} = R_{t+1} + G_{t+1:h}$ $t \le h \le T$
h = horizon = $t+n$

- if $\pi(a|s) = 0$ for a chosen by b
$V(s) = 0$, resulting in high variance

+ instead:

$$G_{t:h} \doteq \rho_t (R_{t+1} + \gamma G_{t+1:h}) + (1-\rho) V_{h-1}(S_t)$$

and $G_{h:h} = V_{h-1}(S_h)$

- when behavior chooses action that target would never choose, DO NOT update V, just ignore sample

+ second term called control variate
+ use with conventional n-step TD learning equation
- action values, a little different

$$G_{t:h} \doteq R_{t+1} + \gamma \left(\rho_{t+1} G_{t+1:h} + \bar{V}_{h-1}(S_{t+1}) - \rho_{t+1} Q_{h-1}(S_{t+1}, A_{t+1}) \right)$$

$$\doteq R_{t+1} + \gamma \rho_{t+1} \left(G_{t+1:h} - Q_{h-1}(S_{t+1}, A_{t+1}) \right) + \gamma \bar{V}_{h-1}(S_t)$$

ends with $G_{h:h} = Q_{h-1}(S_h, A_h)$
if $h > T$, $G_{T-1:h} \doteq R_t$

- analogous to expected SARSA when combined with learning step

- off-policy learning with importance sampling is good, but slow and must be controlled for high variance

+ Can be improved using control variates, autostep step-size in accordance with variance, invariant updates

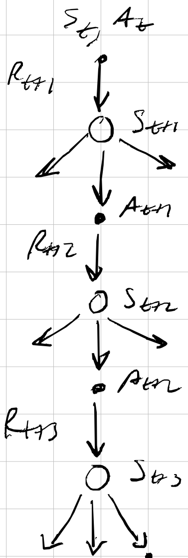
Off Policy Learning Without Importance Sampling: The n-step Tree Backup Algorithm

- Backup example on right for ref
  central spine are sampled S, A, R
- target previously was made up of only sampled data

+ now target will include bootstrapped (estimated) value of actions NOT taken

× only considers <u>leaf</u> nodes

× actions actually taken are not considered



x weight actions by likelihood down the chain

x $\pi(a|S_{t+1})$ for 1st level
$\pi(A_{t+1}|S_{t+1})\,\pi(a'|S_{t+2})$ for 2nd leve
$\pi(A_{t+1}|S_{t+1})\,\pi(A_{t+2}|S_{t+2})\,\pi(a''|S_{t+3})$

- 3-step tree backing is more like 6 "half" steps

+ Sample action to state & from state to all possible actions in policy

- n-step tree backup equation

+ Start with 1-step return = expected SARSA

$$G_{t:t+1} = R_{t+1} + \gamma \sum_a \pi(a|S_{t+1})\, Q_t(S_{t+1}, a) \qquad t < T-1$$

+ 2-step:

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

+ General recursion

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

$$+ \gamma \pi(A_{t+1}, S_{t+1}) G_{t+1:t+n}$$

for $t < T-1$, $n \geq 2$. $G_{T-1:t+n} \doteq R_T$

- combine with general update step

$$Q_{t+n}(S_t, A_t) \doteq Q_{t+n-1}(S_t, A_t) + \alpha$$

$$[G_{t:t+n} - Q_{t+n-1}(S_t, A_t)]$$

for $0 \leq t < T$, & other values are unchanged

$$Q_{t+n}(s,a) = Q_{t+n-1}(s,a) \quad s \neq S_t \; a \neq A_t$$

EXERCISE 7.11 Show that

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

$$\delta_t \doteq R_{t+1} + \gamma \bar{V}_t(S_{t+1}) - Q(S_t, A_t)$$

$$\delta_t = R_{t+1} + \gamma \sum_a \pi(a \mid S_{t+1})\, Q(S_{t+1}, a) - Q(S_t, A_t)$$

$$G_{t:t+n} = R_{t+1} + \gamma \left(\bar{V}_t(S_{t+1}) - Q(A_{t+1}, S_{t+1}) \right)$$

$$\ldots + \gamma\, \pi(A_{t+1}, S_{t+1}) \left(R_{t+2} + \gamma \left(\bar{V}_{t+1}(S_{t+2}) - Q(A_{t+2}, S_{t+2}) \right) \right) + \gamma\, \pi(A_{t+2}, S_{t+2}) \left(R_{t+3} \ldots \right)$$

$$= R_{t+1} + \gamma V_t(S_{t+1}) - \gamma Q(A_{t+1}, S_{t+1}) + \gamma \pi(A_{t+1}, S_{t+1}) R_{t+2} + \gamma^2 \pi\, A_{t+1}, S_{t+1}\, V_{t+1}(S_{t+2}) - \gamma^2 \pi(A_{t+1}, S_{t+1})\, Q(A_{t+2}, S_{t+2}) + \ldots$$

①

$$= ① + Q(S_t, A_t) - Q(S_t, A_t)$$

$$= \left(R_{t+1} + \gamma V_t(S_{t+1}) - Q(S_t, A_t) \right)(1) + \left(R_{t+2} + \gamma V_{t+1}(S_{t+2}) - Q(S_{t+1}, A_{t+1}) \right)\left(\gamma \pi(A_{t+1}, S_{t+1}) \right) + \vdots$$

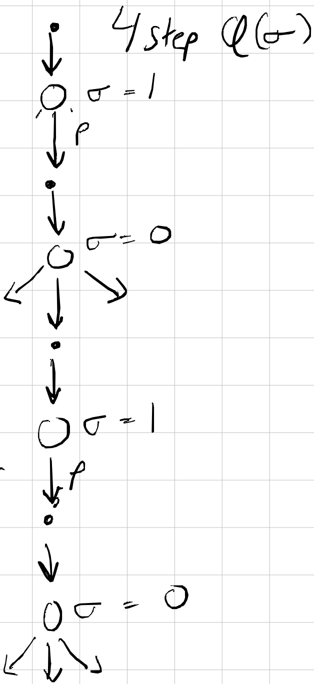
$$= Q(S_t, A_t) + \sum_{k=t}^{\min(t+n-1,\, T-1)} \delta_k \prod_{i=t+1}^{k} \gamma\, \pi(A_i, S_i)$$

*A Unifying Algorithm
n-step Q(σ)

- how do we take n-step SARSA, expected SARSA, & tree backup, all together?

- $\sigma = \boxed{\text{degree of sampling}}$

+ $\sigma = 1$ full sample
  $\sigma = 0$ full expectation, ignore sample



- first, tree backup with t+n=h

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

$$= R_{t+1} + \gamma \bar{V}_{h-1}(S_{t+1}) -$$

$$\gamma\pi(A_{t+1}|S_{t+1})\, Q_{h-1}(S_{t+1}, A_{t+1}) + \gamma\pi(A_{t+1}|S_{t+1})\, G_{t+1:h}$$

$$= R_{t+1} + \gamma\pi(A_{t+1}|S_{t+1})\left(G_{t+1:h} - Q_{h-1}(S_{t+1}, A_{t+1})\right)$$

$$+ \gamma \bar{V}_{h-1}(S_{t+1})$$

- then, slide linearly between σ

$$G_{t:h} \doteq R_{t+1} + \gamma\left(\sigma_{t+1}\rho_{t+1} + 1 - \sigma_{t+1}\pi(A_{t+1}|S_{t+1})\right)$$

$$\left(G_{t+1:h} - Q_{h-1}(S_{t+1}, A_{t+1})\right) + \gamma\bar{V}_{h-1}(S_{t+1})$$

$t \le h \le T$, $G_{h:h} = Q_{h-1}(S_h, A_h)$
if $h < T$, or $G_{T-1:T} \doteq R_T$ if $h = T$

- combine with n-step SARSA, but leave out importance sampling, since it is in return itself.

Summary

- n-step: more computation, more memory
faster learning
+ can be reduced with eligibility traces later