November 11, 2022

7 Exercises

7.1 Question

In Chapter 6 we noted that the Monte Carlo error can be written as the sum of TD errors (6.6) if the value estimates don't change from step to step. Show that the n-step error used in (7.2) can also be written as a sum TD errors (again if the value estimates don't change) generalizing the earlier result.

Answer

Value estimates are assumed not to change thus we can omit value estimate subscripts such that $V_t(S_t) = V_{t+1}(S_t)$.

n-step TD Error used in 7.2 is:

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

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

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

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

$G_{t:t+n} - V(S_t) = \sum_{k=t}^{t+n-1} \gamma^{k-t}\delta_k$

7.2 Question

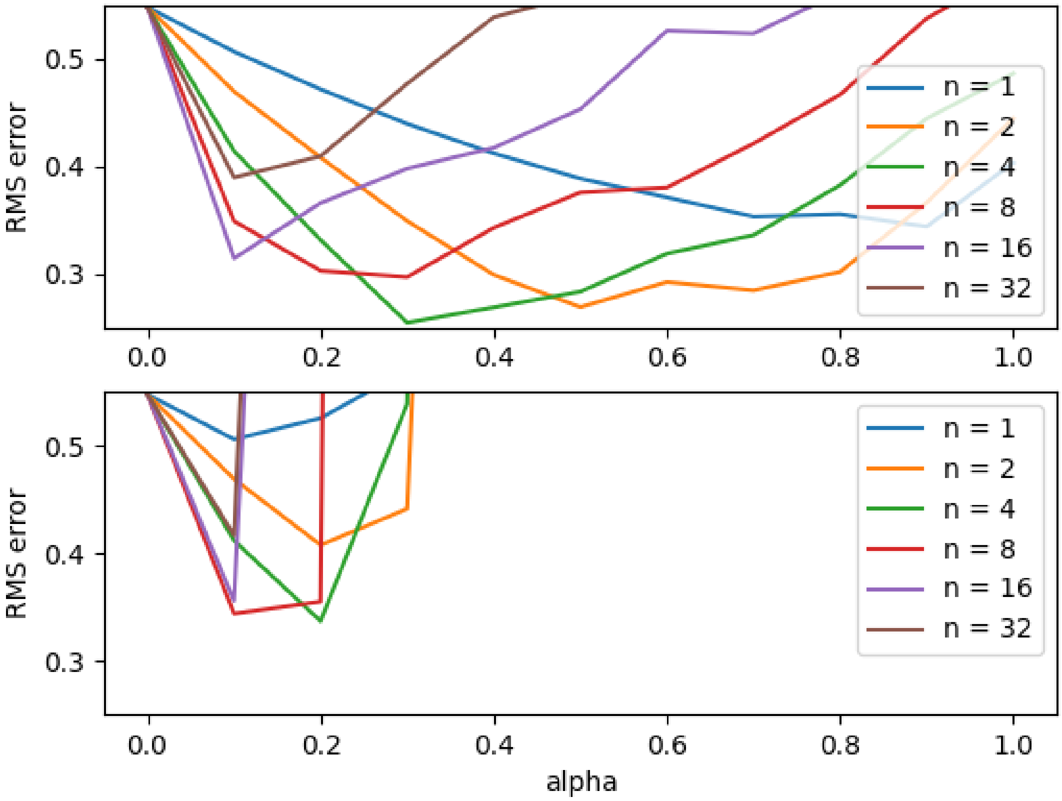
(programming) With an n-step method, the value estimates do change from step to step, so an algorithm that used the sum of TD errors (see previous exercise) in place of the error in (7.2) would actually be a slightly different algorithm. Would it be a better algorithm or a worse one? Devise and program a small experiment to answer this question empirically.

Answer

The chart above shows regular n-step TD with different n parameters. The chart below shows the same configuration with unchanged value functions.

Value function updates are applied only after an episode terminates.

Using sum of TD errors as in place of the error in 7.2 performs worse in all n and α values.



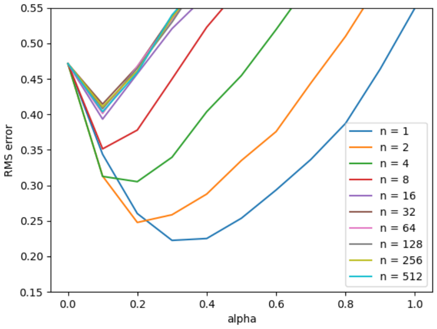
7.3 Question

Why do you think a larger random walk task (19 states instead of 5) was used in the examples of this chapter? Would a smaller walk have shifted the advantage to a different value of n? How about the change in left-side outcome from 0 to -1 made in the larger walk? Do you think that made any difference in the best value of n?

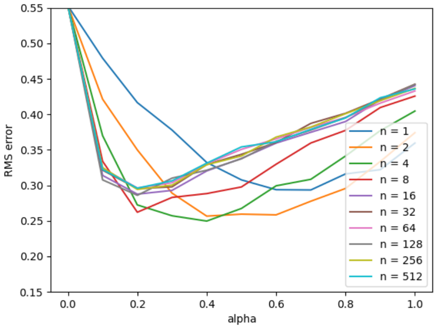
Answer

If n-step size is close to or bigger than the average number steps to complete an episode then the algorithm approaches to MC which involves variance. Using 19 states increases average number of steps to complete an episode thus helps to show how n-step size effects the algorithm.

If number of states was 5, optimum n-step size would be smaller. An empiric study shows that if return value -1 is used with 5 states, most optimum n value would be 1.



Randomwalk with 5 states and reward of 0 on the left, results are found to be different from the -1 case. We can interpolate and conclude that changing the return value may affect the result.



7.4 Question

Prove that the n-step return of Sarsa (7.4) can be written exactly in terms of a novel TD error.

Answer

Given expression:
$G_{t:t+n} = Q_{t-1}(S_t, A_t) + \sum_{k=t}^{\min(t+n,T)-1} \gamma^{k-t}[R_{k+1} + \gamma Q_k(S_{k+1}, A_{k+1}) - Q_{k-1}(S_k, A_k)]$

Can be expanded for n:

$$\begin{aligned}
G_{t:t+n} = Q_{t-1}(S_t, A_t) &+ \gamma^0[R_{t+1} + \gamma Q_t \quad (S_{t+1}, A_{t+1}) - Q_{t-1}(S_t, A_t)] \\
&+\gamma^1[R_{t+2} + \gamma Q_{t+1}(S_{t+2}, A_{t+2}) - Q_t \; (S_{t+1}, A_{t+1})] \\
&+\gamma^2[R_{t+3} + \gamma Q_{t+2}(S_{t+3}, A_{t+3}) - Q_{t+1}(S_{t+2}, A_{t+2})] \\
\dots & \\
&+\gamma^{n-1}[R_{t+n} + \gamma Q_{t+n-1}(S_{t+n}, A_{t+n}) - Q_{t+n-2}(S_{t+n-2}, A_{t+n-2})]
\end{aligned}$$

γ distributed:

$$\begin{aligned}
G_{t:t+n} = Q_{t-1}(S_t, A_t) &+ \quad R_{t+1} + \gamma Q_t \quad (S_{t+1}, A_{t+1}) - Q_{t-1}(S_t, A_t) \\
&+\gamma \; R_{t+2} + \gamma^2 Q_{t+1}(S_{t+2}, A_{t+2}) - \gamma Q_t(S_{t+1}, A_{t+1})] \\
&+\gamma^2 R_{t+3} + \gamma^3 Q_{t+2}(S_{t+3}, A_{t+3}) - \gamma^2 Q_{t+1}(S_{t+2}, A_{t+2})] \\
\dots & \\
&+\gamma^{n-1} R_{t+n} + \gamma^n Q_{t+n-1}(S_{t+n}, A_{t+n}) - \gamma^{n-1} Q_{t+n-2}(S_{t+n-2}, A_{t+n-2})]
\end{aligned}$$

After γ distribution diagonal $\gamma Q_k(S_{k+1}, A_{k+1})$ and $Q_{k-1}(S_k, A_k)$ terms cancel out.
$G_{t:t+n} = Q_{t-1}(S_t, A_t) + R_{t+1} - Q_{t-1}(S_t, A_t) + \gamma^{n-1} R_{t+n} + \gamma^n Q_{t+n-1}(S_{t+n}, A_{t+n})$
$G_{t:t+n} = R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^{n-1} R_{t+n} + \gamma^n Q_{t+n-1}(S_{t+n}, A_{t+n})$

Finally we obtain n-step return of Sarsa (7.4), proved.

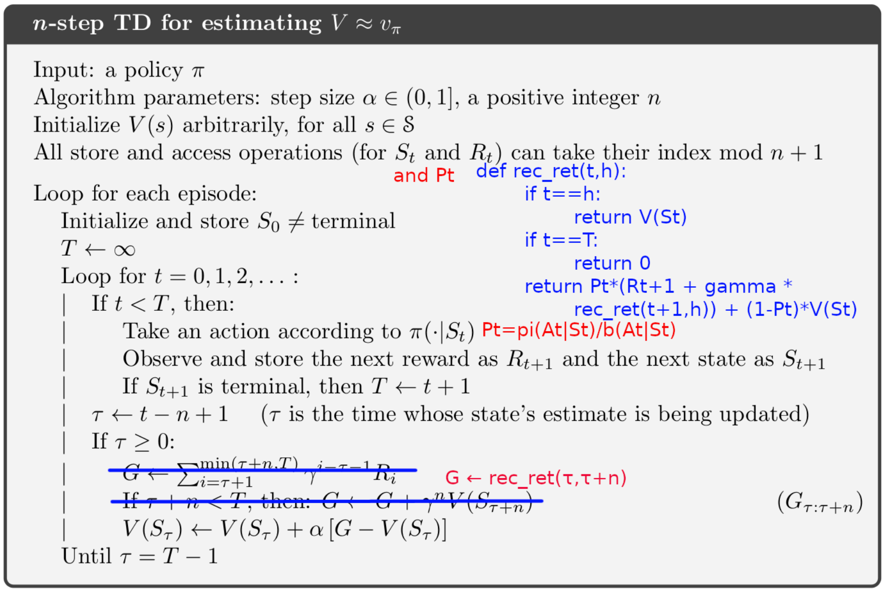
7.5 Question

Write the pseudocode for the off-policy state-value prediction algorithm described above.

Answer

The pseudo-code for ”n-step TD for estimating” modified which reflects equation 7.2.

Return is calculated as defined in equation 7.13.



```
$n$-step TD for estimating $V \approx v_\pi$

Input: a policy $\pi$
Algorithm parameters: step size $\alpha \in (0,1]$, a positive integer $n$
Initialize $V(s)$ arbitrarily, for all $s \in \mathcal{S}$
All store and access operations (for $S_t$ and $R_t$) can take their index mod $n+1$
                                                     and Pt   def rec_ret(t,h):
Loop for each episode:                                            if t==h:
    Initialize and store $S_0 \neq$ terminal                          return V(St)
    $T \leftarrow \infty$                                         if t==T:
    Loop for $t = 0, 1, 2, \ldots$ :                                  return 0
    |   If $t < T$, then:                                         return Pt*(Rt+1 + gamma *
    |       Take an action according to $\pi(\cdot|S_t)$ Pt=pi(At|St)/b(At|St)   rec_ret(t+1,h)) + (1-Pt)*V(St)
    |       Observe and store the next reward as $R_{t+1}$ and the next state as $S_{t+1}$
    |       If $S_{t+1}$ is terminal, then $T \leftarrow t+1$
    |   $\tau \leftarrow t - n + 1$   ($\tau$ is the time whose state's estimate is being updated)
    |   If $\tau \geq 0$:
    |       ~~$G \leftarrow \sum_{i=\tau+1}^{\min(\tau+n,T)} \gamma^{i-\tau-1} R_i$~~     G ← rec_ret(τ,τ+n)
    |       ~~If $\tau + n < T$, then: $G \leftarrow G + \gamma^n V(S_{\tau+n})$~~          $(G_{\tau:\tau+n})$
    |       $V(S_\tau) \leftarrow V(S_\tau) + \alpha [G - V(S_\tau)]$
    Until $\tau = T - 1$
```

7.6 Question

Prove that the control variate in the above equations does not change the expected value of the return.

Answer

We have $E[\rho_t] = E[\frac{\pi(A_t|S_t)}{b(A_t|S_t)}] = 1$ from 5.13.
Equation 7.14:
$G_{t:h} = R_{t+1}\gamma\rho_{t+1}(G_{t+1:h} - Q_{h-1}(S_{t+1}, A_{t+1}) + \gamma\bar{V}_{h-1}(S_{t+1}))$
The expected value is:
$E[G_{t:h}] = E[R_{t+1} + \gamma\rho_{t+1}(G_{t+1:h} - Q_{h-1}(S_{t+1}, A_{t+1})) + \gamma\bar{V}_{h-1}(S_{t+1})]$
Using Linearity of expectation and 5.13:
$E[G_{t:h}] = E[R_{t+1}] + \gamma E[(G_{t+1:h} - Q_{h-1}(S_{t+1}, A_{t+1}))] + \gamma E[\bar{V}_{h-1}(S_{t+1})]$
$E[G_{t:h}] = E[R_{t+1}] + \gamma E[G_{t+1:h}] - \gamma E[Q_{h-1}(S_{t+1}, A_{t+1})] + \gamma E[\bar{V}_{h-1}(S_{t+1})]$
Using 7.8:
$E[G_{t:h}] = E[R_{t+1}] + \gamma E[G_{t+1:h}] - \gamma E[\bar{V}_{h-1}(S_{t+1})] + \gamma E[\bar{V}_{h-1}(S_{t+1})]$
$E[G_{t:h}] = E[R_{t+1}] + \gamma E[G_{t+1:h}]$
$E[G_{t:h}] = E[R_{t+1} + \gamma G_{t+1:h}]$
Using 7.12:
$E[G_{t:h}] = E[G_{t:h}]$

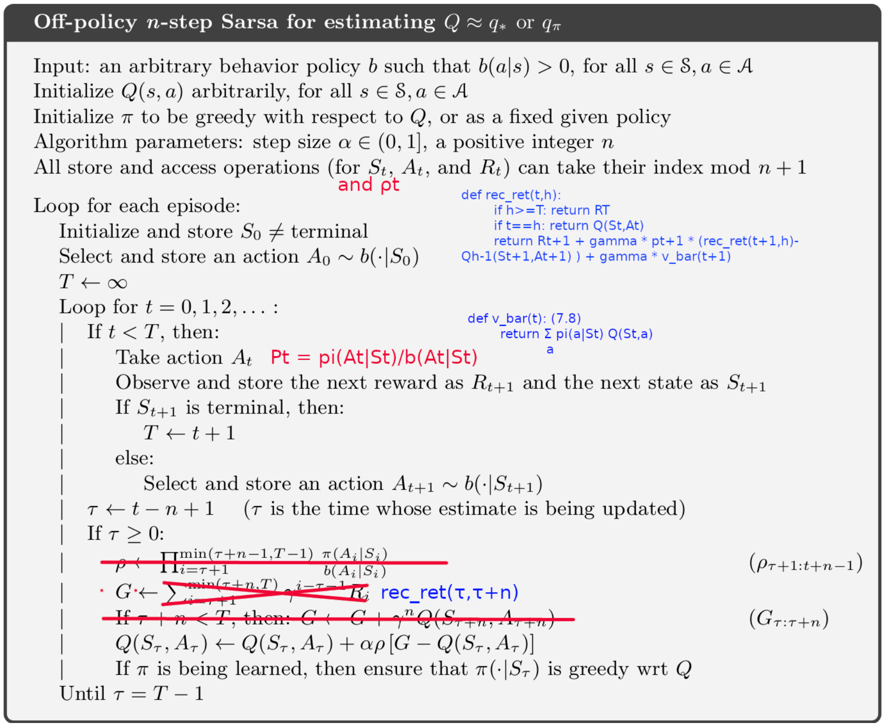
7.7 Question

Write the pseudocode for the off-policy action-value prediction algorithm described immediately above.

Answer

The pseudo-code for ”n-step sarsa for estimating” modified which reflects equation 7.11.

Return is calculated as defined in equation 7.14.



```
Off-policy n-step Sarsa for estimating $Q \approx q_*$ or $q_\pi$

Input: an arbitrary behavior policy $b$ such that $b(a|s) > 0$, for all $s \in \mathcal{S}, a \in \mathcal{A}$
Initialize $Q(s,a)$ arbitrarily, for all $s \in \mathcal{S}, a \in \mathcal{A}$
Initialize $\pi$ to be greedy with respect to $Q$, or as a fixed given policy
Algorithm parameters: step size $\alpha \in (0,1]$, a positive integer $n$
All store and access operations (for $S_t$, $A_t$, and $R_t$) can take their index mod $n+1$
                                 and pt

                                          def rec_ret(t,h):
                                                 if h>=T: return RT
                                                 if t==h: return Q(St,At)
                                                 return Rt+1 + gamma * pt+1 * (rec_ret(t+1,h)-
                                          Qh-1(St+1,At+1) ) + gamma * v_bar(t+1)

Loop for each episode:
   Initialize and store $S_0 \neq$ terminal
   Select and store an action $A_0 \sim b(\cdot|S_0)$
   $T \leftarrow \infty$
   Loop for $t = 0, 1, 2, \ldots$ :
   |  If $t < T$, then:
   |      Take action $A_t$   Pt = pi(At|St)/b(At|St)
   |      Observe and store the next reward as $R_{t+1}$ and the next state as $S_{t+1}$
   |      If $S_{t+1}$ is terminal, then:
   |          $T \leftarrow t+1$
   |      else:
   |          Select and store an action $A_{t+1} \sim b(\cdot|S_{t+1})$
   |  $\tau \leftarrow t-n+1$    ($\tau$ is the time whose estimate is being updated)
   |  If $\tau \geq 0$:
   |      ~~$\rho \leftarrow \prod_{i=\tau+1}^{\min(\tau+n-1,T-1)} \frac{\pi(A_i|S_i)}{b(A_i|S_i)}$~~                ($\rho_{\tau+1:t+n-1}$)
   |      $G \leftarrow$ ~~$\sum_{i=\tau+1}^{\min(\tau+n,T)} \gamma^{i-\tau-1} R_i$~~ rec_ret(τ,τ+n)
   |      ~~If $\tau + n < T$, then: $G \leftarrow G + \gamma^n Q(S_{\tau+n}, A_{\tau+n})$~~          ($G_{\tau:\tau+n}$)
   |      $Q(S_\tau, A_\tau) \leftarrow Q(S_\tau, A_\tau) + \alpha\rho\,[G - Q(S_\tau, A_\tau)]$
   |      If $\pi$ is being learned, then ensure that $\pi(\cdot|S_\tau)$ is greedy wrt $Q$
   Until $\tau = T-1$

                                           def v_bar(t): (7.8)
                                                  return Σ pi(a|St) Q(St,a)
                                                         a
```

7.8 Question

Show that the general (off-policy) version of the n-step return (7.13) can still be written exactly and compactly as the sum of state-based TD errors (6.5) if the approximate state value function does not change.

Answer

Equation 7.13:

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

Assuming state value function does not change.
$G_{t:h} = \rho_t(R_{t+1} + \gamma G_{t+1:h}) + (1 - \rho_t)V(S_t)$
$G_{t:h} = \rho_t(R_{t+1} + \gamma G_{t+1:h} - V(S_t)) + V(S_t)$
$G_{t:h} = \rho_t(R_{t+1} + \gamma G_{t+1:h} - V(S_t) + \gamma V(S_{t+1}) - \gamma V(S_{t+1})) + V(S_t)$
$G_{t:h} = \rho_t(R_{t+1} + \gamma V(S_{t+1}) - V(S_t) + \gamma G_{t+1:h} - \gamma V(S_{t+1})) + V(S_t)$
$G_{t:h} = \rho_t(\delta_t + \gamma G_{t+1:h} - \gamma V(S_{t+1})) + V(S_t)$
$G_{t:h} = \rho_t(\delta_t + \gamma(G_{t+1:h} - V(S_{t+1}))) + V(S_t)$
$G_{t:h} = \rho_t(\delta_t + \gamma(\rho_{t+1}(R_{t+2} + \gamma G_{t+2:h} - V(S_{t+1})) + V(S_{t+1}) - V(S_{t+1}))) + V(S_t)$
$G_{t:h} = \rho_t(\delta_t + \gamma(\rho_{t+1}(R_{t+2} + \gamma G_{t+2:h} - V(S_{t+1})))) + V(S_t)$
$G_{t:h} = \rho_t\delta_t + \gamma\rho_{t:t+1}\delta_{t+1} + \cdots + \gamma^{h-1}\rho_{t:h}(G_{h:h} - V(S_h)) + V(S_t)$
$G_{t:h} = V(S_t) + \sum_{k=t}^{h-1} \rho_{t:k}\gamma^{k-t}\delta_k$

7.9 Question

Repeat the above exercise for the action version of the off-policy n-step return (7.14) and the Expected Sarsa TD error (the quantity in brackets in Equation 6.9).

Answer

Expected Sarsa TD error (the quantity in brackets in Equation 6.9):
$\delta_t = R_{t+1} + \gamma\bar{V}(S_{t+1}) - Q(S_t, A_t)$
Equation 7.14:
$G_{t:h} = R_{t+1} + \gamma\rho_{t+1}(G_{t+1:h} - Q(S_{t+1}, A_{t+1})) + \gamma\bar{V}(S_{t+1})$
$G_{t:h} = R_{t+1} + \gamma\bar{V}(S_{t+1}) - Q(S_t, A_t) + \gamma\rho_{t+1}(G_{t+1:h} - Q(S_{t+1}, A_{t+1})) + Q(S_t, A_t)$
$G_{t:h} = \delta_t + \gamma\rho_{t+1}(G_{t+1:h} - Q(S_{t+1}, A_{t+1})) + Q(S_t, A_t)$
$G_{t:h} = \delta_t + \gamma\rho_{t+1}([R_{t+2} + \gamma\rho_{t+2}(G_{t+2:h} - Q(S_{t+2}, A_{t+2})) + \gamma\bar{V}(S_{t+2})] - Q(S_{t+1}, A_{t+1})) + Q(S_t, A_t)$
$G_{t:h} = \delta_t + \gamma\rho_{t+1}(\delta_{t+1} + \gamma\rho_{t+2}(G_{t+2:h} - Q(S_{t+2}, A_{t+2})) + Q(S_{t+1}, A_{t+1}) - Q(S_{t+1}, A_{t+1})) + Q(S_t, A_t)$
$G_{t:h} = \delta_t + \gamma\rho_{t+1}(\delta_{t+1} + \gamma\rho_{t+2}(G_{t+2:h} - Q(S_{t+2}, A_{t+2}))) + Q(S_t, A_t)$
$G_{t:h} = \delta_t + \gamma\rho_{t+1}\delta_{t+1} + \gamma^2\rho_{t+1:t+2}(G_{t+2:h} - Q(S_{t+2}, A_{t+2})) + Q(S_t, A_t)$
$G_{t:h} = \delta_t + \gamma\rho_{t+1}\delta_{t+1} + \gamma^2\rho_{t+1:t+2}(\delta_{t+2} + G_{t+3:h} - Q(S_{t+3}, A_{t+3})) + Q(S_t, A_t)$
$G_{t:h} = \delta_t + \gamma\rho_{t+1}\delta_{t+1} + \gamma^2\rho_{t+1:t+2}\delta_{t+2} + \gamma^2\rho_{t+1:t+2}(G_{t+3:h} - Q(S_{t+3}, A_{t+3})) + Q(S_t, A_t)$
$G_{t:h} = \delta_t + \gamma\rho_{t+1}\delta_{t+1} + \gamma^2\rho_{t+1:t+2}\delta_{t+2} + \cdots + \gamma^{h-t-2}\rho_{t+1:h-2}\delta_{h-2} + \gamma^{h-t-1}\rho_{t+1:h-1}(G_{h:h} - Q(S_h, A_h)) + Q(S_t, A_t)$
since $G_{h:h} = Q(S_h, A_h)$
$G_{t:h} = \delta_t + \gamma\rho_{t+1}\delta_{t+1} + \gamma^2\rho_{t+1:t+2}\delta_{t+2} + \cdots + \gamma^{h-t-2}\rho_{t+1:h-2}\delta_{h-2} + Q(S_t, A_t)$

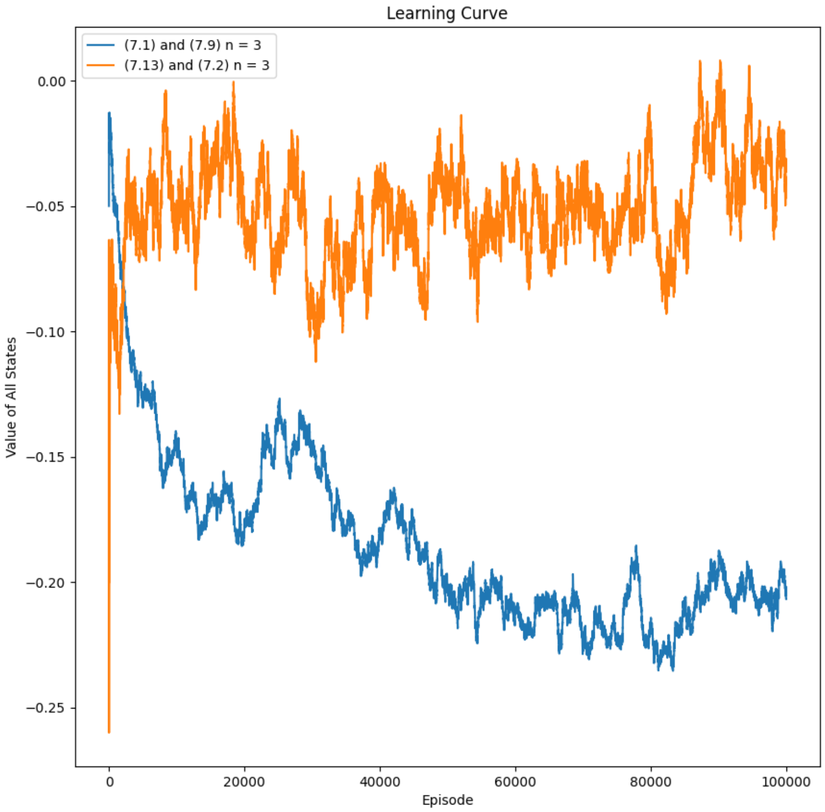
$G_{t:h} = Q(S_t, A_t) + \sum_{k=t}^{h-2} \gamma^{k-t}\rho_{t+1:k}\delta_k$ (given $\rho_{t:h} = 1$ if $t > h$)

7.10 Question

(programming) Devise a small off-policy prediction problem and use it to show that the off-policy learning algorithm using (7.13) and (7.2) is more data efficient than the simpler algorithm using (7.1) and (7.9).

Answer

Data efficiency is measured by looking at how fast state values converges. All state values are summed and averaged. The chart below shows average state value at episode for the simple and sophisticated approaches.



My experimental finding is that: The sophisticated approach suffers from the control variate. When an action is optimal and not favored by the random policy (e.g. $\rho_t = 1.0/0.5 = 2$), resulting importance sampling ratio signifies not only the return term but also the control variate. This makes the returns fluctuate. This may be due to an implementation or setup error.

However, the simpler TD seems to work better thus more data efficient.