Chapter 7 solutions

Hanna Gábor

1. *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.*

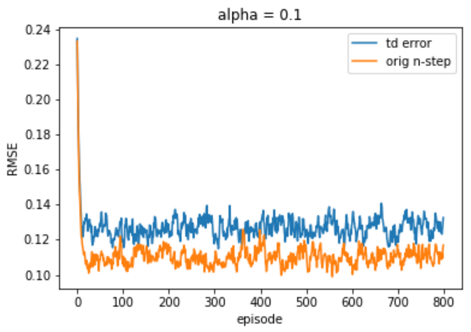
$$\begin{aligned}
G_{t:t+n} - V(S_t) &= R_{t+1} + \gamma G_{t+1:t+n} - V(S_t) + \gamma V(S_{t+1}) - \gamma V(S_{t+1}) \\
&= \delta_t + \gamma (G_{t+1:t+n} - V(S_{t+1})) \\
&= \delta_t + \gamma \delta_{t+1} + \gamma^2 \delta_{t+2} + \cdots + \gamma^{n-1} \delta_{t+n-1} + \gamma^n V(S_{t+n}) - \gamma^n V(S_{t+n}) \\
&= \sum_{k=t}^{t+n-1} \gamma^{k-t} \delta_k
\end{aligned}$$

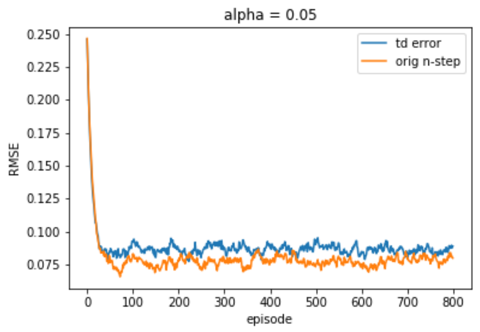
2. *Programming. With an n-step method, the value estimates do change from step to step, so an algorithm that used the sum of TD errors (see previous exercise) in place of the error in (7.2) would actually be a slightly different algorithm. Would it be a better algorithm or a worse one? Devise and program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

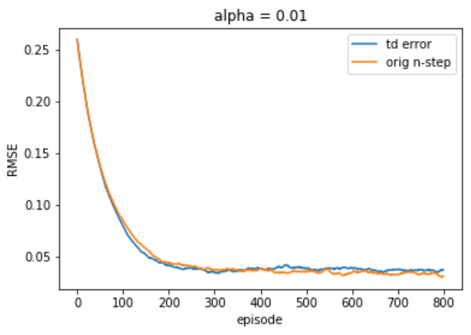
   There is only a difference in the two methods if we update state estimes for some states from $\{S_{t+1}, \ldots S_{t+n}\}$ between time steps t and $t + n$. Because of this reason, I would like to try the algorithms on a problem where we may return to the same state shortly. My expectation is that the n-step update works better, because it uses a more recent estimation of $V(S_t)$. It does not use estimations of $V_(S_{t+i})$ (for $i = 1 \cdots < n$) either, but I am not sure if it is for the better or the worse. Let's try it on a random walk example, like in Example 6.2.

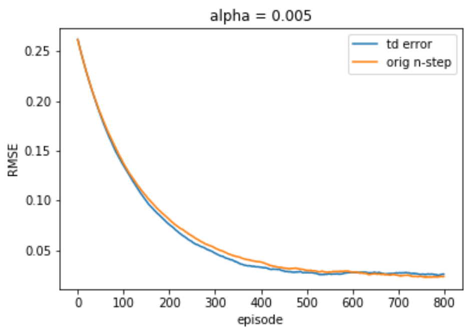
   The code can be found at https://github.com/hannagabor/SBRL/blob/master/7.2/random_walks_td.ipynb.

   The results show that if alpha is bigger, then the TD-error update is worse than the original n-step method. If alpha is small enough, then there is no difference. (At least on this problem.)









3. *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?*

   If n is big, but the random walk has only a few states, then there is a higher chance that we update states that are far from the end state. That sounds bad.

   As for the left-side outcome: I can't see any change.

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

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

$$\begin{aligned}
G_{t:t+n} &= R_{t+1} + \gamma G_{t+1:t+n} + Q_{t-1}(S_t, A_t) - Q_{t-1}(S_t, A_t) \\
&+ \gamma Q_t(S_{t+1}, A_{t+1}) - \gamma Q_t(S_{t+1}, A_{t+1}) \\
&= Q_{t-1}(S_t, A_t) + (R_{t+1} + \gamma Q_t(S_{t+1}, A_{t+1}) - Q_{t-1}(S_t, A_t)) \\
&+ \gamma(G_{t+1:t+n} - Q_t(S_{t+1}, A_{t+1})) \\
&+ \gamma^2 Q_{t+1}(S_{t+2}, A_{t+2}) - \gamma^2 Q_{t+1}(S_{t+2}, A_{t+2}) \\
&= Q_{t-1}(S_t, A_t) + (R_{t+1} + \gamma Q_t(S_{t+1}, A_{t+1}) - Q_{t-1}(S_t, A_t)) \\
&+ \gamma(R_{t+2} + \gamma^2 Q_{t+1}(S_{t+2}, A_{t+2}) - Q_t(S_{t+1}, A_{t+1})) \\
&+ \gamma^2(G_{t+2:t+n} - Q_{t+1}(S_{t+2}, A_{t+2})) \\
&+ \gamma^3 Q_{t+2}(S_{t+3}, A_{t+3}) - \gamma^2 Q_{t+2}(S_{t+3}, A_{t+3}) \\
&\dots \\
&= Q_{t-1}(S_t, A_t) + \sum_{k=t}^{min(t+n,T)-1} \gamma^{k-t}(R_{k+1} + \gamma Q_k(S_{k+1}, A_{k+1}) - Q_{k-1}(S_k, A_k))
\end{aligned}$$

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

   The update rule we want to use is the following.

$$V_{t+n}(S_t) = V_{t+n-1}(S_t) + \alpha\Big(\sum_{k=t}^{t+n-1} \gamma^{k-t}\big(\rho_{t:k}R_{k+1} + \rho_{t:k-1}(1-\rho_k)V_{t+n-1}(S_k)\big) \\ + \gamma^n \rho_{t:t+n}V_{t+n-1}(S_{t+n}) - V_{t+n-1}(S_t)\Big)$$

See the algorithm on the next page.

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

   We need to show that

$$\mathbb{E}(\bar{V}_{h-1}(S_{t+1}) - \rho_{t+1}Q_{h-1}(S_{t+1}, A_{t+1})) = 0$$

   $\bar{V}_{h-1}(S_{t+1})$ does not depend on b, so it is enough to show that

$$\sum_a b(a|S_{t+1})\Big(\frac{\pi(a|S_{t+1})}{b(a|S_{t+1})}Q_{h-1}(S_{t+1}, a)\Big) = \bar{V}_{h-1}(S_{t+1})$$

   The left-hand side is equal to $\sum_a \pi(a|S_{t+1})Q_{h-1}(S_{t+1}, a)$ and that is the definition of $\bar{V}_{h-1}(S_{t+1})$.

Algorithm 1 Per-decision off-policy state-value prediction algorithm

```
Inputs: a target policy $\pi$ and a behavior policy $b$ such that
$b(a|s) > 0$ if $\pi(a|s) > 0$.
Initialize $V(s) \in \mathbb{R}$, arbitrarily, for all $s \in S$.
Algorithm parameters: steps size $\alpha \in (0, 1]$, a positive integer $n$.
All store and access operations (for $S_t$ and $R_t$) can take their index mod $n + 1$.
for each episode do
  Initialize and store $S_0$ non-terminal state.
  $T \leftarrow \infty$
  for $t = 0, 1, 2 \ldots$ do
    if $t < T$ then
      Take an action according to $b$.
      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
        $T = t + 1$
      end if
    end if
    $\tau = t - n + 1$ ($\tau$ is the time whose state's estimate is being updated.)
    if $\tau \geq 0$ then
      $G \leftarrow 0$
      $\rho \leftarrow 1$
      for $k = \tau, \ldots, min(\tau + n - 1, T - 1)$ do
        $\rho_k = \frac{\pi(A_k|S_k)}{b(A_k|S_k)}$
        $G \leftarrow G + \gamma^{k-\tau}\rho(1 - \rho_k)V(S_k)$
        $\rho \leftarrow \rho \cdot \rho_k$
        $G \leftarrow G + \gamma^{k-\tau}\rho R_{k+1}$
      end for
      if $\tau + n < T$ then
        $\rho \leftarrow \rho \cdot \frac{\pi(A_{\tau+n}|S_{\tau+n})}{b(A_{\tau+n}|S_{\tau+n})}$
        $G \leftarrow G + \gamma^n V(S_{\tau+n})$
      end if
      $V(S_\tau) \leftarrow V(S_\tau) + \alpha(G - V(S_\tau))$
    end if
    if $\tau = T - 1$ then
      break
    end if
  end for
end for
```