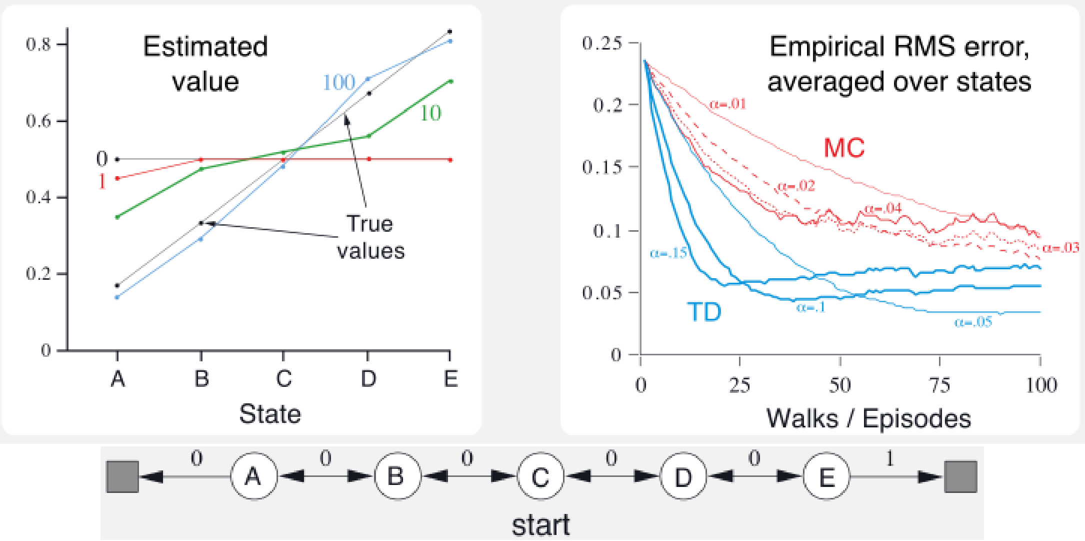
CSE – 426 Homework 11

Griffin Kent

1) Exercise 6.3 of the RL book. Just explain why when using the first episode, the TD update of the value function v will occur on $v(A)$ only.

Exercise 6.3) From the results shown in the left graph of the random walk example it appears that the first episode results in a change in only $V(A)$. What does this tell you about what happened on the first episode? Why was only the estimate for this one state changed? By exactly how much was it changed?



(Solution): In this problem, we are given $\gamma = 1$ because the problem is undiscounted, $\alpha = 0.1$, and all the state-value functions were initialized to the intermediate value $V(s) = 0.5$. We are also given that every episode begins at state C. For each step in a given episode, the $TD(0)$ algorithm performs the following updates:

$$V(S) \leftarrow V(S) + \alpha[R + \gamma V(S') - V(S)]$$
$$S \leftarrow S'.$$

Since the $TD(0)$ algorithm updates $V(S)$ only based on the current state S and the next state S' in any given episode E, we can conclude that for $V(A)$ to be the only change in episode one that the episode must have terminated on the left-side terminal state with a reward of 0. To give an example of this, let's say that the first episode looked like the following:

$$E = \{C, left, 0, B, left, 0, A, left, 0\}.$$

Then the $TD(0)$ algorithm would have performed the following updates:

$$V(C) \leftarrow V(C) + \alpha[R + \gamma V(B) - V(C)] = 0.5 + 0.1(0 + 0.5 - 0.5) = 0.5.$$
$$V(B) \leftarrow V(B) + \alpha[R + \gamma V(A) - V(B)] = 0.5 + 0.1(0 + 0.5 - 0.5) = 0.5.$$
$$V(A) \leftarrow V(A) + \alpha[R + \gamma 0 - V(A)] = 0.5 + 0.1(0 + 0 - 0.5) = 0.45.$$

Therefore, we can see that only $V(A)$ has changed to 0.45 and all other state-values remain at 0.5. Notice that the exact sequence of transitions in the first episode make no difference, as long as the episode terminated on the left-side.

■

2) Suppose you use the linear function $\boldsymbol{w}^T\boldsymbol{x}(s,a)$ to approximate the action-value function $q_\pi(s,a)$. Give the stochastic semi-gradient descent equation for $\boldsymbol{w}$ when using TD for policy evaluation. Then explain why this is not the regular stochastic gradient descent update.

(Solution): The stochastic semi-gradient descent equation for $\boldsymbol{w}$ when using TD for policy evaluation is defined as follows (Eq. (86) if the lecture notes):

$$\boldsymbol{w} \leftarrow \boldsymbol{w} + \alpha[R_{t+1} + \gamma\hat{q}_\pi(S_{t+1}, A_{t+1}; \boldsymbol{w}) - \hat{q}_\pi(S_t, A_t; \boldsymbol{w})]\nabla_w\hat{q}_\pi(S_t, A_t; \boldsymbol{w}).$$

If the linear function $\boldsymbol{w}^T\boldsymbol{x}(s,a)$ is used to approximate $q_\pi(s,a)$, then the update for $\boldsymbol{w}$ can be written as

$$\boldsymbol{w} \leftarrow \boldsymbol{w} + \alpha[R_{t+1} + \gamma\boldsymbol{w}^T\boldsymbol{x}(S_{t+1}, A_{t+1}) - \boldsymbol{w}^T\boldsymbol{x}(S_t, A_t)]\boldsymbol{x}(S_t, A_t).$$

This stochastic semi-gradient descent update is not regular stochastic gradient descent because the target $R_{t+1} + \gamma\boldsymbol{w}^T\boldsymbol{x}(S_{t+1}, A_{t+1})$ depends on the parameter $\boldsymbol{w}$ whereas in the regular stochastic gradient descent update, the target is G_t and is free of $\boldsymbol{w}$. Since the update for stochastic semi-gradient descent depends on the current value of $\boldsymbol{w}$, the output will be biased and will not yield a true gradient-descent method. True gradient methods require that the target be independent of $\boldsymbol{w}$.

■

3) Exercise 6.1 of the RL book.

Exercise 6.1) If V changes during the episode, then (6.6) only holds approximately; what would the difference be between the two sides? Let V_t denote the array of state values used at time t in the TD error (6.5) and in the TD update (6.2). Redo the derivation above to determine the additional amount that must be added to the sum of TD errors in order to equal the Monte Carlo error.

(Solution): If we let $V_t \doteq V(S_t)$ and $V_{t+1} \doteq V(S_{t+1})$, then we can redefine the TD error as the following:

$$\begin{aligned} \delta_t &\doteq R_{t+1} + \gamma V_{t+1} - V_t \\ &= R_{t+1} + \gamma V_{t+1} - (V_t + \alpha[R_t + \gamma V_{t+1} - V_t]) \\ &= R_{t+1} + \gamma V_{t+1} - V_t - \alpha[R_t + \gamma V_{t+1} - V_t]. \end{aligned}$$

Then, we can derive the Monte Carlo error when V changes as

$$\begin{aligned} G_t - V_t &= R_{t+1} + \gamma G_{t+1} - V_t \\ &= \delta_t + \gamma(G_{t+1} - V_{t+1}) - \alpha[R_t + \gamma V_{t+1} - V_t] \\ &= \delta_t - \alpha[R_t + \gamma V_{t+1} - V_t] + \gamma\delta_{t+1} - \gamma\alpha[R_{t+1} + \gamma V_{t+2} - V_{t+1}] + \gamma^2(G_{t+2} - V_{t+2}) \\ &= \delta_t - \alpha[R_t + \gamma V_{t+1} - V_t] + \gamma\delta_{t+1} - \gamma\alpha[R_{t+1} + \gamma V_{t+2} - V_{t+1}] + \cdots + \gamma^{T-t-1}\delta_{T-1} \\ &\quad - \gamma^{T-t-1}\alpha[R_{T-1} + \gamma V_T - V_{T-1}] + \gamma^{T-t}(G_T - V_T) \end{aligned}$$

Since $(G_T - V_T) = 0$, this can be written in the following summation form:

$$= \sum_{k=t}^{T-1} \gamma^{k-t}\delta_k - \alpha\sum_{k=t}^{T-1}\gamma^{k-t}[R_k + \gamma V_{k+1} - V_k] = \sum_{k=t}^{T-1}\gamma^{k-t}(\delta_k - \alpha\delta_{k+1}).$$

■

4) Exercise 5.1 of the RL book.

Exercise 5.1) Consider the diagrams on the right in Figure 5.1. Why does the estimated value function jump up for the last two rows in the rear? Why does it drop off for the whole last row on the left? Why are the frontmost values higher in the upper diagrams than in the lower?

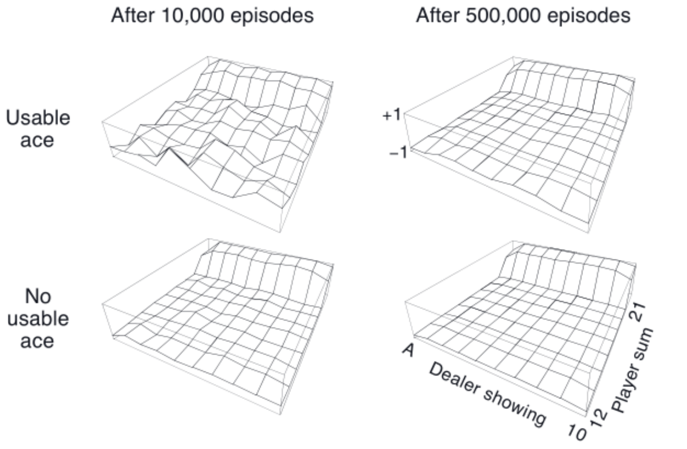


Figure 5.1: Approximate state-value functions for the blackjack policy that sticks only on 20 or 21, computed by Monte Carlo policy evaluation. ■

(Solution):

In this problem, it is stated that the player's policy is to stick if the player's sum is 20 or 21, otherwise they take another card. The player will also get a $+1$ reward if they win and a -1 reward if they lose. In the last two rows in the figures, the player's sum is at 20 or 21, indicating that they will stick. The reason these two rows jump up is because this is the point at which the player has a very high probability of winning and thus the estimated value is closer to $+1$.

The last row on the left indicates that the dealer is showing an ace. The reason that this last row drops off is because regardless of what the player's sum is, the dealer has more opportunities to get closer to 21 without busting. Therefore, the estimated value is slightly lower as it indicates that there is a more likely probability that the player may not win.

The frontmost values in the upper diagram are higher in the upper diagrams than in the lower for a similar reason for why the leftmost row is lower for the player. In the frontmost values in the upper diagrams, the player has a usable ace. This means that the player's current sum is 12 and their potential ways of getting to 20 or 21 in a single draw are to obtain the following cards: $\{9,8,10\}$. However, in the lower diagrams, the only way that the player can get to 20 or 21 is if they draw one of the following: $\{9,8\}$. Thus, since in the upper diagrams the player has more opportunities to win than in the lower diagrams, the estimated value is slightly higher in the upper.

■

5) (Graduate Only) Given a trajectory $\{s_0, a_0, r_1, s_1, a_1, r_2, s_2, a_2, 0\}$, with $\boldsymbol{x}(s_0, a_0) = [1,1]^T$, $\boldsymbol{x}(s_1, a_1) = [2,2]^T$, $\boldsymbol{x}(s_2, a_2) = [3,3]^T$, $r_1 = 0$, and $r_2 = 1$. First modify the "Semi-gradient TD(0) for estimating $\hat{v} \approx v_\pi$" algorithm on Page 203 of the RL book (Section 9.3) to estimate $\hat{q} \approx q_\pi$. (No need to give the full algorithm but just point out the modification you made to the original algorithm.) Assuming that $\hat{q}(s, a) = \boldsymbol{w}^T \boldsymbol{x}(s, a)$, show the first two update steps of $\boldsymbol{w}$ using the modified stochastic semi-gradient descent algorithm, based on the above trajectory.
Parameters: The learning rate α and the discounting factor γ can be assumed to be known and you don't need to give explicit values to them.
[*Hints: Which* (s, a) *pairs will be involved in the update? What's the target of the update?*]

(Solution): The only step that needs to be changed in the algorithm "Semi-gradient TD(0) for estimating $\hat{v} \approx v_\pi$" is the update step. We derived the update for $\boldsymbol{w}$ using the stochastic semi-gradient when using TD for policy evaluation with the linear function approximation for the action-value function $q_\pi(s, a)$ in problem (2) above. Thus, we only need to replace the update step for $\boldsymbol{w}$ in the algorithm on page 203 with the following:

$$\boldsymbol{w} \leftarrow \boldsymbol{w} + \alpha[R + \gamma \hat{q}_\pi(S', A'; \boldsymbol{w}) - \hat{q}_\pi(S, A; \boldsymbol{w})]\nabla_w \hat{q}_\pi(S, A; \boldsymbol{w}).$$

Since in this problem we will be using the same linear approximation $\hat{q}(s, a) = \boldsymbol{w}^T \boldsymbol{x}(s, a)$, we will be using the update:

$$\boldsymbol{w} \leftarrow \boldsymbol{w} + \alpha[R_{t+1} + \gamma \boldsymbol{w}^T \boldsymbol{x}(S_{t+1}, A_{t+1}) - \boldsymbol{w}^T \boldsymbol{x}(S_t, A_t)]\boldsymbol{x}(S_t, A_t).$$

Given a starting value $\boldsymbol{w_0}$**,** the first update will be:

$$\boldsymbol{w_1} \leftarrow \boldsymbol{w_0} + \alpha[R_1 + \gamma \boldsymbol{w}_0^T \boldsymbol{x}(S_1, A_1) - \boldsymbol{w}_0^T \boldsymbol{x}(S_0, A_0)]\boldsymbol{x}(S_0, A_0)$$
$$= \boldsymbol{w_0} + \alpha\left[\gamma \boldsymbol{w}_0^T \begin{bmatrix} 2 \\ 2 \end{bmatrix} - \boldsymbol{w}_0^T \begin{bmatrix} 1 \\ 1 \end{bmatrix}\right]\begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

Similarly, the second update will be:

$$\boldsymbol{w_2} \leftarrow \boldsymbol{w_1} + \alpha[R_2 + \gamma \boldsymbol{w}_1^T \boldsymbol{x}(S_2, A_2) - \boldsymbol{w}_1^T \boldsymbol{x}(S_1, A_1)]\boldsymbol{x}(S_1, A_1)$$
$$= \boldsymbol{w_1} + \alpha\left[1 + \gamma \boldsymbol{w}_1^T \begin{bmatrix} 3 \\ 3 \end{bmatrix} - \boldsymbol{w}_1^T \begin{bmatrix} 2 \\ 2 \end{bmatrix}\right]\begin{bmatrix} 2 \\ 2 \end{bmatrix}.$$

■