Reinforcement Learning: Tutorial 6

Advanced TD methods

Week 3
University of Amsterdam

Milena Kapralova

September 2024

Check-in

- How is it going?
- How is HW2?
- Are you ready to start HW3?
- If you have any feedback so far, please mail me at *m.kapralova@uva.nl*

Outline

1. Admin
2. Advanced TD exercises
3. Ask anything about HW2 or 5.2+5.3 (HW3)

Admin

- Reminder that HW2 deadline is tomorrow @ 17:00
- Any questions?

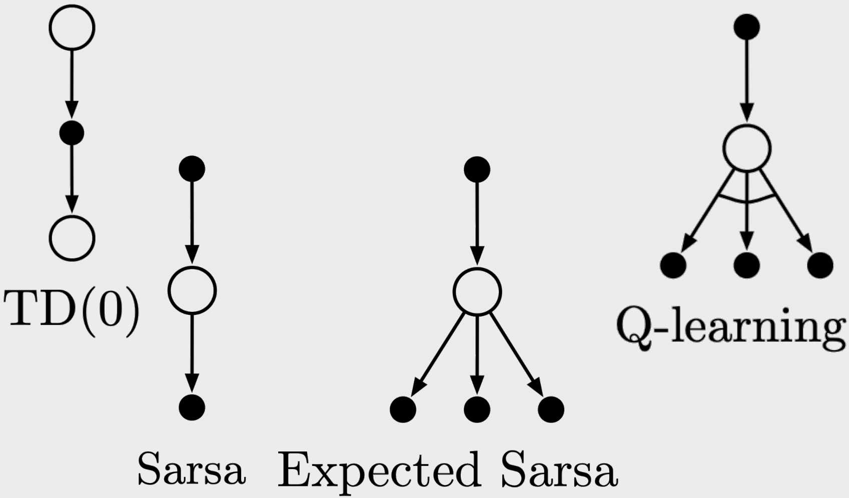
Tutorial 6 Overview

1. Advanced TD exercises
2. Ask anything about HW2 or 5.2+5.3 (HW3)

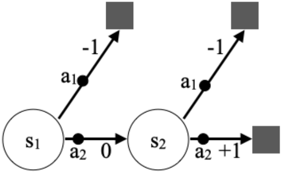
Tutorial 6 Overview

1. Advanced TD exercises
   - Question 5.1
2. Ask anything about HW2 or 5.2+5.3 (HW3)

Theory Intermezzo: TD(0), SARSA, E SARSA, Q-learning

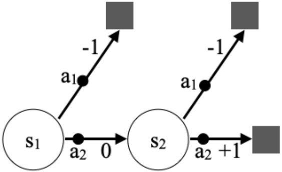


Q 5.1 Off-policy TD



- MDP, the undiscounted case ($\gamma = 1$)
- uniform behavior policy b ($p(a_1) = p(a_2) = 0.5$ in both states
- target policy π: $p(a_1) = 0.1$ and $p(a_2) = 0.9$ in both states

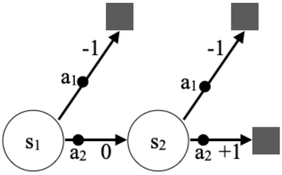
Q 5.1 Off-policy TD



- MDP, the undiscounted case ($\gamma = 1$)
- uniform behavior policy b ($p(a_1) = p(a_2) = 0.5$ in both states
- target policy π: $p(a_1) = 0.1$ and $p(a_2) = 0.9$ in both states

1. What are the Q functions Q^b and Q^π under both policies?

Q 5.1 Off-policy TD



- MDP, the undiscounted case ($\gamma = 1$)
- uniform behavior policy b ($p(a_1) = p(a_2) = 0.5$ in both states
- target policy π: $p(a_1) = 0.1$ and $p(a_2) = 0.9$ in both states

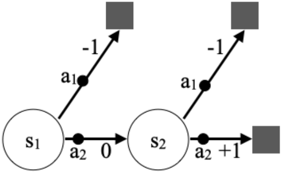
1. What are the Q functions Q^b and Q^π under both policies?

   To calculate the Q functions, perform value iteration. It is the easiest to start at the terminal states and work your way back. For either Q function, we immediately find $Q(s_1, a_1) = -1, Q(s_2, a_1) = -1, Q(s_2, a_2) = +1$. Now, we can calculate $Q(s_1, a_2)$ using the Bellman operator:

$$Q^b(s_1, a_2) = 0.5 \cdot Q(s_2, a_1) + 0.5 \cdot Q(s_2, a_2) = 0$$

$$Q^\pi(s_1, a_2) = 0.1 \cdot Q(s_2, a_1) + 0.9 \cdot Q(s_2, a_2) = 0.8$$

Q 5.1 Off-policy TD



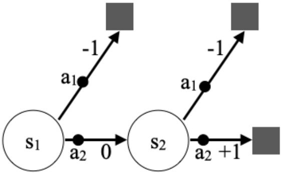
- MDP, the undiscounted case ($\gamma = 1$)
- uniform behavior policy b ($p(a_1) = p(a_2) = 0.5$ in both states
- target policy π: $p(a_1) = 0.1$ and $p(a_2) = 0.9$ in both states

Now consider a dataset gathered using b $(s_1, a_2, 0, s_2, a_1, -1), (s_1, a_2, 0, s_2, a_2, +1)$. Consider a Q-function that is initialized as per the table:

	a_1	a_2
s_1	-1	0.5
s_2	-1	+1

2. Apply one pass of SARSA on the dataset with a learning rate of 0.1. How does the change in Q function relate to the two functions you calculated in sub-question 1?
Hint: throughout this question, only $Q(s_1, a_2)$ will change. Why?

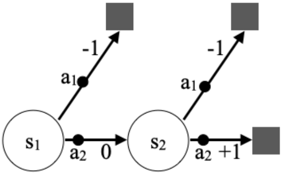
Q 5.1 Off-policy TD



- MDP, the undiscounted case ($\gamma = 1$)
- uniform behavior policy b ($p(a_1) = p(a_2) = 0.5$ in both states
- target policy π: $p(a_1) = 0.1$ and $p(a_2) = 0.9$ in both states

Let's start with the hint. $Q(s_1, a_1), Q(s_2, a_1), Q(s_2, a_2)$ will have targets that are equal to their current values, so they will not change. So we only look at $Q(s_1, a_2)$. The first trajectory results in an update with a target of $r + Q(s_2, a_1) = -1$. So the update is $0.1(-1 - 0.5) = -0.15$, bringing $Q(s_1, a_2)$ to $0.5 - 0.15 = 0.35$. The second trajectory has a target of $r + Q(s_2, a_2) = +1$. The second update will thus be $0.1(1 - 0.35) = 0.065$, bringing $Q(s_1, a_2)$ to $0.35 + 0.065 = 0.415$. In aggregate, this on-policy update changed the Q-function in the direction of Q^b. Repeated passes would converge Q^b.

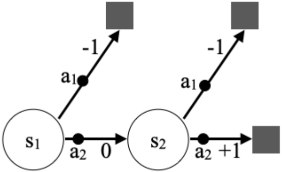
Q 5.1 Off-policy TD



- MDP, the undiscounted case ($\gamma = 1$)
- uniform behavior policy b ($p(a_1) = p(a_2) = 0.5$ in both states
- target policy π: $p(a_1) = 0.1$ and $p(a_2) = 0.9$ in both states

3. We can use expected SARSA to estimate the Q-function Q^{π}. Apply a single pass, and note how the change in Q function relates to the two functions of sub-question 1.

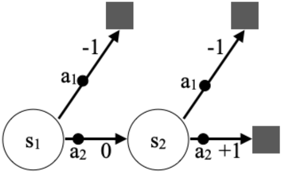
Q 5.1 Off-policy TD



- MDP, the undiscounted case ($\gamma = 1$)
- uniform behavior policy b ($p(a_1) = p(a_2) = 0.5$ in both states
- target policy π: $p(a_1) = 0.1$ and $p(a_2) = 0.9$ in both states

Again, we only look at $Q(s_1, a_2)$. The expected SARSA update doesn't look at action taken in the following time step, as the target is $r + E_{a\sim\pi(s')}Q(s', a) = 0.9 * +1 + 0.1 * -1 = 0.8$.
So the first update is $0.1(0.8 - 0.5) = 0.03$, resulting in $Q(s_1, a_2) = 0.5 + 0.03 = 0.53$. The second update is $0.1(0.8 - 0.53) = 0.027$, resulting in $Q(s_1, a_2) = 0.53 + 0.027 = 0.557$.
In aggregate, this off-policy update changed the Q-function in the direction of Q^π. Repeated passes would converge to Q^π.

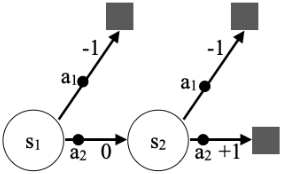
Q 5.1 Off-policy TD



- MDP, the undiscounted case ($\gamma = 1$)
- uniform behavior policy b ($p(a_1) = p(a_2) = 0.5$ in both states
- target policy π: $p(a_1) = 0.1$ and $p(a_2) = 0.9$ in both states

4. Another possibility for off-policy learning is applying SARSA with importance weight. Again do one pass and notice the change in Q-function.

Q 5.1 Off-policy TD

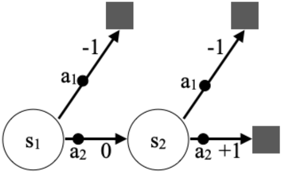


- MDP, the undiscounted case ($\gamma = 1$)
- uniform behavior policy b ($p(a_1) = p(a_2) = 0.5$ in both states
- target policy π: $p(a_1) = 0.1$ and $p(a_2) = 0.9$ in both states

Now, in the first update we have a target of -1 like in the on-policy case, but with an importance weight of $\pi(a_1|s_2)/b(a_1|s_2) = 0.2$. This results in an update of $0.1 * 0.2 * (-1 - 0.5) = -0.03$, so $Q(s_1, a_2) = 0.5 - 0.03 = 0.47$. In the second update, we have a target of $+1$, with an importance weight of $\pi(a_2|s_2)/b(a_2|s_2) = 1.8$. This results in an update of $0.1 * 1.8 * (1 - 0.47) \approx 0.095$, so $Q(s_1, a_2) \leftarrow 0.47 + 0.095 = 0.565$.

Again, in aggregate, this off-policy update changed the Q-function in the direction of Q^π. Repeated passes would converge to Q^π.

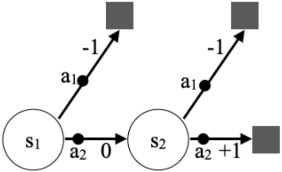
Q 5.1 Off-policy TD



- MDP, the undiscounted case ($\gamma = 1$)
- uniform behavior policy b ($p(a_1) = p(a_2) = 0.5$ in both states
- target policy π: $p(a_1) = 0.1$ and $p(a_2) = 0.9$ in both states

5. What is the main difference in the outcomes of expected SARSA and SARSA with importance weights? Which should we prefer?

Q 5.1 Off-policy TD



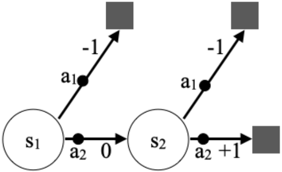
- MDP, the undiscounted case ($\gamma = 1$)
- uniform behavior policy b ($p(a_1) = p(a_2) = 0.5$ in both states
- target policy π: $p(a_1) = 0.1$ and $p(a_2) = 0.9$ in both states

5. What is the main difference in the outcomes of expected SARSA and SARSA with importance weights? Which should we prefer?

Both variants in principle converge to the same Q function. However, importance weights add a lot of variance. This can make learning unstable unless a small learning rate is set (in which case learning becomes slow).

So, expected SARSA is the preferred option.

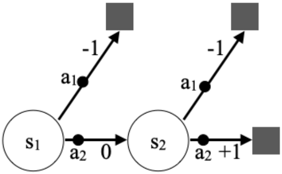
Q 5.1 Off-policy TD



- MDP, the undiscounted case ($\gamma = 1$)
- uniform behavior policy b ($p(a_1) = p(a_2) = 0.5$ in both states
- target policy π: $p(a_1) = 0.1$ and $p(a_2) = 0.9$ in both states

6. We could also learn a V function, e.g. through TD(0), off policy. For example, by using importance weights. Why do you think the book doesn't cover this?

Q 5.1 Off-policy TD

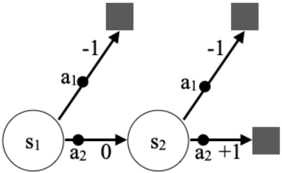


- MDP, the undiscounted case ($\gamma = 1$)
- uniform behavior policy b ($p(a_1) = p(a_2) = 0.5$ in both states
- target policy π: $p(a_1) = 0.1$ and $p(a_2) = 0.9$ in both states

6. We could also learn a V function, e.g. through TD(0), off policy. For example, by using importance weights. Why do you think the book doesn't cover this?

This is certainly possible. However, in policy evaluation, we are often interested in evaluating the behavior policy, so off-policy learning is not as relevant. In control, where we are learning a policy, policies change over time, and we have the exploration vs. exploitation issue. Both of these issues make off-policy learning much more important.

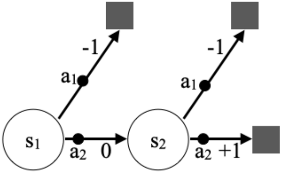
Q 5.1 Off-policy TD



- MDP, the undiscounted case ($\gamma = 1$)
- uniform behavior policy b ($p(a_1) = p(a_2) = 0.5$ in both states
- target policy π: $p(a_1) = 0.1$ and $p(a_2) = 0.9$ in both states

7. Could you do something like expected Sarsa for learning a V function? If yes, apply one pass. If not, explain why this is the case.

Q 5.1 Off-policy TD

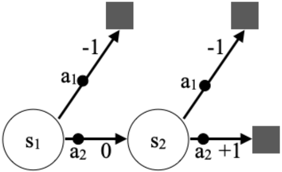


- MDP, the undiscounted case ($\gamma = 1$)
- uniform behavior policy b ($p(a_1) = p(a_2) = 0.5$ in both states
- target policy π: $p(a_1) = 0.1$ and $p(a_2) = 0.9$ in both states

7. Could you do something like expected Sarsa for learning a V function? If yes, apply one pass. If not, explain why this is the case.

For the update of $V(s_1)$, we would need to take the expectation over the 'targets' from all possible actions. We typically do not have that information. In this particular example, the dataset doesn't have any information about the result of applying a_1 from this state, for example. This is the same reason as why we cannot use something like expected SARSA to Monte-Carlo learning: it requires data about the effects of trying actions that we typically do not have access to.

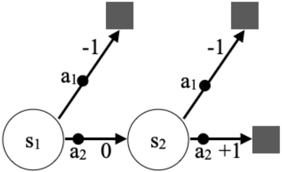
Q 5.1 Off-policy TD



- MDP, the undiscounted case ($\gamma = 1$)
- uniform behavior policy b ($p(a_1) = p(a_2) = 0.5$ in both states
- target policy π: $p(a_1) = 0.1$ and $p(a_2) = 0.9$ in both states

8. Why is Q-learning considered an off-policy control method? (Ex. 6.11 in RL:AI)

Q 5.1 Off-policy TD



- MDP, the undiscounted case ($\gamma = 1$)
- uniform behavior policy b ($p(a_1) = p(a_2) = 0.5$ in both states
- target policy π: $p(a_1) = 0.1$ and $p(a_2) = 0.9$ in both states

8. Why is Q-learning considered an off-policy control method? (Ex. 6.11 in RL:AI)

   In Q-learning the target policy is always greedy w.r.t its current values. However, its behavior policy can be anything e.g. epsilon greedy and continues to visit all state-action pairs during learning. Note that Q-learning is a special case of expected SARSA.

WELL DONE!

Tutorial 6 Overview

1. Advanced TD exercises
2. Ask anything about HW2 or 5.2+5.3 (HW3)
    - Questions 3.4+4.3, 5.2+5.3

Ask anything about HW2 or 5.2+5.3 (HW3)

- 3.4: Coding (+ Little bit of theory)
- 4.3: Theory
- 5.2: Coding (+ Little bit of theory)
- 5.3: Theory

That's it!

Good luck with the HW and see you on Monday!