Prof. Joschka Bödecker, Gabriel Kalweit

Reinforcement Learning
Exercise 3

Before we learn how to use the methods from this week for control (we actually implement this next week), we first have to understand the basic concepts. So, this week is a mix of theory and practice.

0 Lecture

Watch *Lecture 04: Model-free Control*[1] before the upcoming session on Friday, November 16.

1 Monte Carlo and TD(λ)

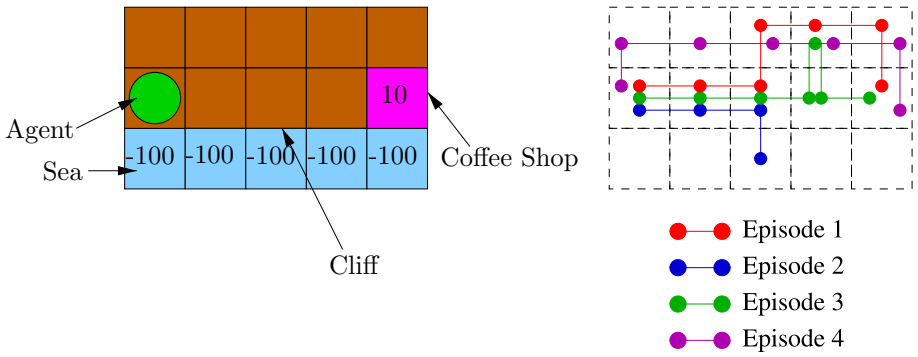

Figure 1: Cliff MDP

Consider the MDP in Figure 1, where all actions (an action moves the agent in a desired direction: up, down, left or right) succeed with a probability of 0.8. With a probability of 0.2 the agent moves randomly in another direction. All transitions result in a reward of -1, except when the coffee shop is reached (terminal state $s_{2,5}$: reward of 10) or if the agent falls of the cliff (terminal states $s_{3,1} \dots s_{3,5}$: reward of -100). The agent always starts in state $s_{2,1}$ as indicated in Figure 1.

(a) Using Monte-Carlo policy evaluation, calculate $V_3(i)$ for all states i based on the illustrated episodes 1 to 3 (right part of Figure 1). Use the first-visit-method, i.e. every state is updated only once – on the first-visit – per episode, even if the state is visited again during the episode. In this task, we estimate the value by a running mean with $\alpha_t = \frac{1}{t}$ for episode t and $V_0(i) = 0$ for all i. We do not discount, i.e. $\gamma = 1$.

[1] `https://ilias.uni-freiburg.de/goto.php?target=xvid_1121348&client_id=unifreiburg`

(b) Consider now Episode 4 (magenta). Specify for all states visited during this episode the TD-error based on the value-function $V_3(\cdot)$ calculated in (a).

(c) Using the TD(λ)-algorithm, determine for $\lambda = 0$, $\lambda = 0.5$ and $\lambda = 1.0$ the expected value $v_\pi(s_{2,1})$ based on the first three episodes. *Hint: In the lecture slides, the TD (λ)-update is defined as* $V(s_t) \leftarrow V(s_t) + \alpha(G_t^\lambda - V(s_t))$, *where* $G_t^\lambda = (1-\lambda)\sum_{n=1}^{\infty} \lambda^{n-1} G_t^{(n)}$ *(in the episodic setting, this becomes* $G_t^\lambda = (1-\lambda)\sum_{n=1}^{T-t-1} \lambda^{n-1} G_t^{(n)} + \lambda^{T-t-1} G_t^{(T-t)}$*). It can be converted to TD-form (so you can also reuse the results from (b)):* $V(s_t) \leftarrow V(s_t) + \alpha \sum_{i=0}^{\infty} \lambda^i \delta_{t+i}$. *You can find the proof in the appendix.*

2 First-visit MC Evaluation

Implement the First-visit MC Evaluation algorithm introduced in the first part of Lecture 4,

`mc_evaluation(policy, env, num_episodes, discount_factor=1.0)`,

in `mc_evaluation.py`, where

- `policy` is a function that maps an observation to action probabilities and
- `env` is an OpenAI gym environment.

It returns a dictionary that maps from state to value.

This task is based on the Blackjack example from the lecture[2] and an implementation can be found in `blackjack.py`. The state is a tuple – containing the players current sum, the dealer's one showing card (1-10 where 1 is ace) and whether or not the player holds a usable ace (0 or 1) – and the value is a float. You find the tests in `exercise-03_test.py`. They expect an average return. Run them by

```
python exercise-03_test.py -v
```

or by

```
python -m unittest exercise-03_test.py -v.
```

In addition, you also find a visualization script of the predicted value-functions for which you need `matplotlib`[3]. You can run it by

```
python mc_evaluation_visualization.py.
```

3 Experiences

Make a post in thread *Week 03: Model-free Prediction* in the forum [4], where you provide a brief summary of your experience with this exercise, the corresponding lecture and the last meeting.

[2] `http://www0.cs.ucl.ac.uk/staff/d.silver/web/Teaching_files/MC-TD.pdf#page=8`
[3] `https://matplotlib.org/users/installing.html`
[4] `https://ilias.uni-freiburg.de/goto.php?target=frm_1121060&client_id=unifreiburg`

A Appendix

Lemma A.1

$$(1-\lambda)\sum_{n=m}^{\infty}\lambda^{n-1}=\lambda^{m-1}$$

Proof.

$$\begin{aligned}
(1-\lambda)\sum_{n=m}^{\infty}\lambda^{n-1} &= (1-\lambda)\left(\sum_{n=1}^{\infty}\lambda^{n-1}-\sum_{k=1}^{m-1}\lambda^{k-1}\right)\\
&= (1-\lambda)\left(\underbrace{\sum_{n=0}^{\infty}\lambda^{n}-\sum_{k=0}^{m-2}\lambda^{k}}_{\text{geometric series}}\right)\\
&= (1-\lambda)\left(\frac{1}{1-\lambda}-\frac{1-\lambda^{m-1}}{1-\lambda}\right)\\
&= 1-(1-\lambda^{m-1})\\
&= \lambda^{m-1}
\end{aligned}$$

□

Lemma A.2

$$\sum_{n=1}^{\infty}\lambda^{n-1}\sum_{m=1}^{n}R_{t+m}=\sum_{m=1}^{\infty}R_{t+m}\sum_{n=m}^{\infty}\lambda^{n-1}$$

Proof.

$$\begin{aligned}
\sum_{n=1}^{\infty}\lambda^{n-1}\sum_{m=1}^{n}R_{t+m} &= \lambda^{0}R_{t+1}+\lambda^{1}(R_{t+1}+R_{t+2})+\lambda^{2}(R_{t+1}+R_{t+2}+R_{t+3})+\cdots\\
&= R_{t+1}(\lambda^{0}+\lambda^{1}+\lambda^{2}+\cdots)+R_{t+2}(\lambda^{1}+\lambda^{2}+\cdots)+R_{t+3}(\lambda^{2}+\cdots)+\cdots\\
&= \sum_{m=1}^{\infty}R_{t+m}\sum_{n=m}^{\infty}\lambda^{n-1}
\end{aligned}$$

□

Lemma A.3

$$(1-\lambda)\sum_{n=1}^{\infty}\lambda^{n-1}\sum_{m=1}^{n}R_{t+m}=\sum_{m=1}^{\infty}\lambda^{m-1}R_{t+m}$$

Proof.

$$\begin{aligned}(1-\lambda)\sum_{n=1}^{\infty}\lambda^{n-1}\sum_{m=1}^{n}R_{t+m} &\overset{2}{=} (1-\lambda)\sum_{m=1}^{\infty}R_{t+m}\sum_{n=m}^{\infty}\lambda^{n-1}\\ &= \sum_{m=1}^{\infty}R_{t+m}(1-\lambda)\sum_{n=m}^{\infty}\lambda^{n-1}\\ &\overset{1}{=} \sum_{m=1}^{\infty}\lambda^{m-1}R_{t+m}\end{aligned}$$

□

Lemma A.4

$$(1-\lambda)\sum_{n=1}^{\infty}\lambda^{n-1}V(S_t+n) = V(S_t) + \sum_{m=1}^{\infty}\lambda^{m-1}V(S_{t+m}) - V(S_{t+m-1})$$

Proof.

$$\begin{aligned}(1-\lambda)\sum_{n=1}^{\infty}\lambda^{n-1}V(S_t+n) &= \sum_{n=1}^{\infty}V(S_t+n)(\lambda^{n-1}-\lambda^n)\\ &= V(S_{t+1})(\lambda^0-\lambda^1) + V(S_{t+2})(\lambda^1-\lambda^2) + V(S_{t+3})(\lambda^2-\lambda^3)+\cdots\\ &= \lambda^0V(S_{t+1}) + \lambda^1V(S_{t+2}) - \lambda^1V(S_{t+1}) + \lambda^2V(S_{t+3}) - \lambda^2V(S_{t+2})\cdots \underbrace{+\lambda^0V(S_t) - \lambda^0V(S_t)}_{\text{add this}}\\ &= V(S_t) + \sum_{m=1}^{\infty}\lambda^{m-1}V(S_{t+m}) - V(S_{t+m-1})\end{aligned}$$

□

Theorem A.5 *The TD(λ)-update is defined as* $V(s_t) \leftarrow V(s_t) + \alpha(G_t^\lambda - V(s_t))$, *where* $G_t^\lambda = (1-\lambda)\sum_{n=1}^{\infty}\lambda^{n-1}G_t^{(n)}$. *It can be converted to TD-form,* $V(s_t) \leftarrow V(s_t) + \alpha\sum_{i=0}^{\infty}\lambda^i\delta_{t+i}$.

Proof. The total return of an episode is defined as $G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots + \gamma^{T-1}R_T$. The value of a state s given policy π is then $v_\pi(s) = \mathbb{E}_\pi[G_t|S_t = s]$. We can estimate this expectation by the empirical mean, an incremental mean or a running mean, i.e. $V(S_t) \leftarrow V(S_t) + \alpha(G_t - V(S_t))$, where α is the learning rate and G_t is the target.
In TD(λ), the target is $G_t^\lambda = (1-\lambda)\sum_{n=1}^{\infty}\lambda^{n-1}G_t^{(n)}$, where $G_t^{(n)} = R_{t+1}+\gamma R_{t+2}+\gamma^2 R_{t+3}+\cdots+\gamma^{n-1}R_{t+n}+\gamma^n V(S_{t+n})$. For simplification, we will ignore the discount.

$$
\begin{aligned}
v(S_t) &= (1-\lambda)\cdot\mathbb{E}\left[\sum_{n=1}^{\infty}\lambda^{n-1}G_t^{(n)}\right] \\
&= (1-\lambda)\cdot\mathbb{E}\left[\sum_{n=1}^{\infty}\lambda^{n-1}\left(\sum_{m=1}^{n}R_{t+m}+V(S_{t+n})\right)\right] \\
&\overset{\frac{3}{4}}{=}\mathbb{E}\left[\sum_{m=1}^{\infty}\lambda^{m-1}R_{t+m}+V(S_{t+m})-V(S_{t+m-1})+V(S_t)\right] \\
&= \mathbb{E}\left[\sum_{m=1}^{\infty}\lambda^{m-1}\delta_{t+m}+V(S_t)\right]
\end{aligned}
$$

If we plug this into our value function update, we get:

$$
V(S_t)\leftarrow V(S_t)+\alpha\left(\sum_{m=1}^{\infty}\lambda^{m-1}\delta_{t+m}+V(S_t)-V(S_t)\right),
$$

which leads to:

$$
V(S_t)\leftarrow V(S_t)+\alpha\sum_{m=0}^{\infty}\lambda^{m}\delta_{t+m+1}.
$$

□