Reinforcement Learning Spring 2023 — Prof. Dr. Paul Swoboda

Exercise Sheet 2 - Due date 23.3.2023, 10am

## Importance Sampling – 8 points

Recall the importance sampling ratio

$$\rho_{k,T} = \frac{\prod_{i=k}^{T} \pi(a_k|s_k)}{\prod_{i=k}^{T} b(a_k|s_k)}$$

for the target policy $\pi$ and the behaviour policy $b$.

a. Prove that unweighted importance sampling is unbiased, i.e.

$$\mathbb{E}_b\left[\frac{\sum_{k\in\mathcal{T}(s_k)} \rho_{k:T(k)} g_k}{|\mathcal{T}(s_k)|}\right] = v_\pi(s_k)$$

where $\mathcal{T}(s_k)$ are all trajectories starting with state $s_k$.

b. Prove that the weighted importance sampling converges, i.e.

$$\left[\frac{\sum_{k\in\mathcal{T}(s_k)} \rho_{k:T(k)} g_k}{\sum_{k\in\mathcal{T}(s_k)} \rho_{k:T(k)}}\right] \to v_\pi(s_k)$$

as $|\mathcal{T}(s_k)| \to \infty$. Note that the underlying distribution from which values are sampled is still $b$. $T(k)$ is the length of the trajectory.

Hint: you can use the strong law of large numbers. It states that for $X_1, X_2 \ldots,$ iid samples then

$$\frac{1}{n}\sum_{i=1}^{n} X_i \to \mu$$

almost surely, that is for almost all realizations except for a zero probability set. Consider suitable augmented numerator and denominator and use the law of large numbers on both of them.

c. Why is the weighted importance sampling biased?

## Preliminaries for Implementing Learning Agents using the gym Environment

We will use two environments to try out this exercises control problems: (i) the easier `FrozenLake-v1` and (ii) the harder `Blackjack-v1` environment.

You can install both environments using `pip install gymnasium` and load them in python using `env = gym.make('FrozenLake-v1')` and `env = gym.make('Blackjack-v1')` respectively. In order to make diagnostics easier, we will use `tqdm` for writing progress bars. Install with `pip` if package is not present in your system.

The following two exercises shall be implemented by completing the given code skeleton.

## MC Control – 8 points

Implement MC control with weighted importance sampling. The behaviour policy should be the $\epsilon$-greedy version of the target policy. Evaluate the algorithm on both environments and generate visualizations for the `FrozenLake` environment. Save the visualization for `FrozenLake` and final greedy performance (you can screen-shot the script output) for both environments and send them in.

## Expected SARSA – 8 points

Implement expected SARSA. Evaluate the algorithm on both environments. Save the visualization for `FrozenLake` and final greedy performance (you can screen-shot the script output) for both environments and send them in.