IN4320 Machine Learning

Exercises: Reinforcement Learning

Author:
Milan Niestijl, 4311728

June 14, 2017

Exercise 1

Claim
The return $R_t = \sum_{h=0}^{\infty} \gamma^h r_{t+h+1}$ is bounded for $0 \leq \gamma < 1$ and bounded rewards $-10 \leq r_{t+h+1} \leq 10$ for all $h \in \mathbb{N}$.

Proof
Using the geometric series, we find:

$$|R_t| \leq \sum_{h=0}^{\infty} |\gamma^h r_{t+h+1}| \leq \sum_{h=0}^{\infty} 10\gamma^h = \frac{10}{1-\gamma} < \infty$$

Exercise 2

The optimal policy for $\gamma = 0.5$, the Q-function is shown after each iteration in the tables 1.

Table 1: Q-value for different iterations.

Action\State	1	2	3	4	5	6
Left	0	0	0	0	0	0
Right	0	0	0	0	0	0

Action\State	1	2	3	4	5	6
Left	0	1	0	0	0	0
Right	0	0	0	0	5	0

Action\State	1	2	3	4	5	6
Left	0	1	0.5	0	0	0
Right	0	0	0	2.5	5	0

Action\State	1	2	3	4	5	6
Left	0	1	0.5	0.625	1.25	0
Right	0	0.625	1.25	2.5	5	0

The resulting optimal policy is shown in table 2.

Table 2: Optimal policy for $\gamma = 0.5$.

s	1	2	3	4	5	6
$\pi(s)$	Right	Left	Right	Right	Right	Right

Exercise 3

The optimal value functions for $\gamma \in \{0, 0.1, 0.9, 1\}$ are shown in table 3. Note that $\gamma = 1$ will work here because the reward function has a finite horizon.

Table 3: $Q*$ for various values of the discount factor γ.

$\gamma = 0$:

Action\State	1	2	3	4	5	6
Left	0	1	0	0	0	0
Right	0	0	0	0	5	0

$\gamma = 0.1$:

Action\State	1	2	3	4	5	6
Left	0	1	0.1	0.01	0.005	0
Right	0	0.01	0.05	0.5	5	0

$\gamma = 0.9$:

Action\State	1	2	3	4	5	6
Left	0	1	3.2805	3.645	4.05	0
Right	0	3.645	4.05	4.5	5	0

$\gamma = 1$:

Action\State	1	2	3	4	5	6
Left	0	1	5	5	5	5
Right	0	5	5	5	5	0

Exercise 4

In figure ..., the 2-norm error of the learned Q-function is plotted versus the number of iterations for several values of the exploration rate ϵ and the learning rate α. It can be seen that for larger values of ϵ, the error decreases faster. This is explained by the fact that more often a random action is taken, which effectively allows the system to gain more new information and alter its values accordingly. Similarly, for larger values of α, the learned Q-function on average updates with larger steps, which is reflected in the figure by the fact that the error makes larger 'jumps'. In this case, this causes the algorithm to converge faster.