Report on Project 3: Black Jack

Ao Wang, 15300240004

May 15, 2018

1 Overview

This project is about solving Black Jack using MDPs. This pdf contains the answers of the required questions.

2 Problem 1

2.1 Problem 1a

As required by the instructions, the state -2 and 2 are the exits, so we manually set the $V_{opt}(s)$ of them to a fixed number, which is 0. Then the results are as follows:

Table 2.1: The result of 1st value iteration (after 0 iterations).

State s	-2	-1	0	1	2
$V_{opt}(s)$	0.0	**11.0**	**-5.0**	**25.0**	0.0

Table 2.2: The result of 2nd value iteration (after 1 iterations).

State s	-2	-1	0	1	2
$V_{opt}(s)$	0.0	**10.0**	**10.2**	**21.5**	0.0

Table 2.3: The result of 3rd value iteration (after 2 iterations).

State s	-2	-1	0	1	2
$V_{opt}(s)$	0.0	**13.04**	**8.45**	**32.14**	0.0

2.2 Problem 1b

After the iteration is converged, the result of π_{opt} is as follows:

Table 2.4: The result of the optimal actions after being converged.

State s	-2	-1	0	1	2
$\pi_{opt}(s)$	-	**-1**	**+1**	**+1**	-

The result is pretty clear because the agent tries to get to the exit as soon as possible as the reward of each step is negative and both of the exits have comparatively high positive rewards. But at state 0, agent will have higher expectation by moving +1, although it has a high possibility of moving -1.

3 Problem 2

3.1 Problem 2a

According to the instruction, the transition (probability) has been changed with noise, with a probability to randomly change to a possible reachable state. We can find a counter example that $V_1(s_{start}) \leq V_2(s_{start})$. We can construct a MDP by making a high probability on going to an exit with low rewards and a low probability on going to an exit with high reward. The simplest situation of this is a MDP with three states. The answer is in the `submission.py`.