Reinforcement Learning

Using Monte Carlo Learning Methods

Assignment 1

In this assignment you will design and build a learning agent that operates in a 6x6 grid world (see Figure 1). Its aim is to get to the goal in the top right-hand corner without crossing any barriers. It should be able to get to the goal no matter where you start it in the grid.

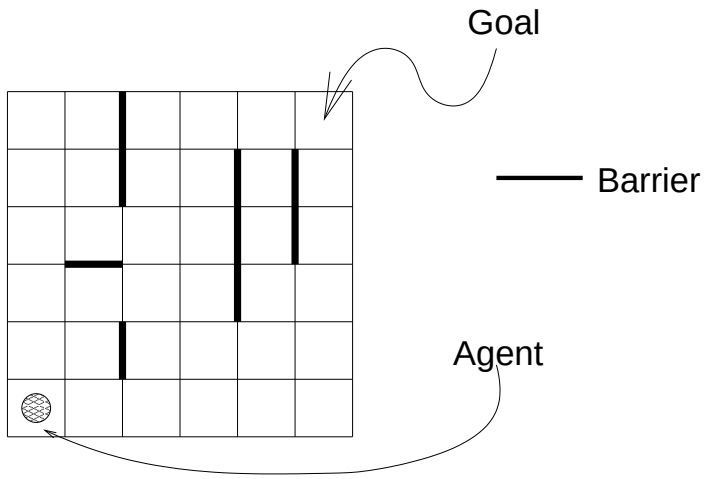

Figure 1: The agent in its 6x6 grid world.

You can assume:

State: that you know what square the agent is in at any one time.

Action: N, S, E, W.

State transitions: deterministic, can't pass through walls. Agent remains on same square if it attempts an action that would take it through the wall or off the grid.

Rewards: -1 for each time step, 0 for reaching the goal.

It is up to you to make decisions about γ.

Answer questions 1, 2 and 3:

1. (a) Figure 2 shows part of a deterministic, non-optimal policy. Complete the policy as you wish: it should be deterministic and must stick to the actions shown in the squares through which the arrow passes (my aim is for you to use a policy that is non-optimal by at least taking the long way round the barriers; but you can make it non-optimal in other parts of the grid too. It should, however, get the agent to the goal). Using Monte Carlo learning,

write a program and carry out experiments to evaluate the value function V^π for the states in the grid under your policy π. Give the value function and your policy.

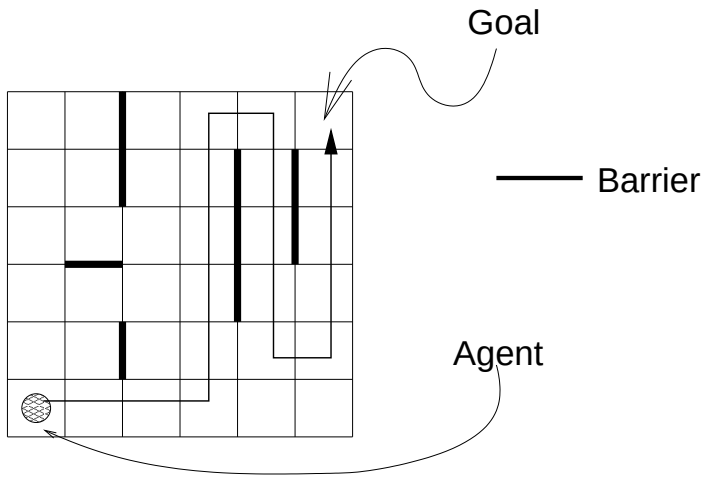

Figure 2: Part of a non-optimal policy.

(b) Describe your experimental method used in 1a. How quickly does your program learn the value function?

(c) Now do Generalised Policy Iteration (GPI), i.e. alternating policy improvement and policy evaluation (still using MC), to get the optimal policy. You can take advantage of your knowledge (a) that the grid world's transition function is deterministic and (b) of the reward function. Give a diagram of the optimal policy your program produces.

(d) Describe your experimental method of 1c and discuss the performance of your program in learning the optimal policy.

2. (a) Choose another deterministic, non-optimal policy (different from those of part 1) and draw a diagram of it. Use Monte Carlo learning to get the Q values of that policy. Give the Q values. Explain what you had to change in your method of 1a to get the Q values.

(b) What are the V values for this policy? Show your working.

(c) Now use the Monte Carlo Exploring Starts algorithm to compute the optimal Q values. Give these Q values.

(d) What are the optimal V values? Show your working.

(e) What is the optimal policy? Show it on a diagram. If you run the experiment again, do you get the same policy? How many optimal policies are there? You don't need to give the absolute number, but describe what they are, using a diagram to illustrate.

3. What are the comparative advantages and disadvantages of using GPI vs. Monte Carlo Exploring Starts to get the optimal policy? Discuss.

You should submit commented code (preferably Matlab for brevity) and a report in which you answer the questions above.

The submission deadline is **Monday 3rd March at 4pm**, i.e. Monday of week 9. The homework is worth 10% of the total course mark. The relative weighting of the three sections is 45%, 40%, 15%.

You need to make TWO forms of submission.

1. Submit your code and a pdf version of your paper electronically using the `submit` program:

```
submit msc rl 1 <filename-of-your-paper> <filename-of-your-code>
```

Please use `msc` even if you are in another year or degree and see the man page for `submit` if you wish to submit more than 2 files. Submit your report as a pdf file and your code as a plain text file or files.

2. ALSO submit your paper to the ITO (you do not need to submit the code to the ITO).

Gillian Hayes
February 7th 2008