HONG, Yu (yh3220)

Imperial College
London

Department of Computing
Academic Year **2020-2021**
Page created Tue Nov 3 23:15:06 GMT 2020

CATE

70051 rac101 2
t5 yh3220 v1

0 0 1 2 0 0 0 1 5 1 7 2

Electronic submission

Mon - 02 Nov 2020
22:03:44

yh3220

Exercise Information

Module:	70051 Introduction to Symbolic Artificial Intelligence (MSc AI)	**Issued:**	Tue - 20 Oct 2020
Exercise:	2 (CW)	**Due:**	Tue - 03 Nov 2020
Title:	Logic	**Assessment:**	Individual
FAO:	Craven, Robert (rac101)	**Submission:**	Electronic

Student Declaration - Version 1

- I declare that this final submitted version is my unaided work.

Signed: (electronic signature) Date: 2020-11-02 21:58:07

For Markers only: (circle appropriate grade)

HONG, Yu (yh3220)	01968443	t5	2020-11-02 21:58:07	**A* A B C D E F**

October 21, 2020

Department of Computing

Reinforcement Learning – Prof Aldo Faisal & Dr Ed Johns

Assessed coursework 1

Version 1.01

To be returned as online submission.

The final document should be submitted in **PDF** format, preferably typeset, ideally in Latex. Your answers should be yours, i.e., written by you, in your own words, showing your own understanding. Written answers should be clear, complete and concise. Figures should be clearly readable, labelled and visible. Poorly produced answers, irrelevant text not addressing the point and unclear text may loose points. As a rule of thumb if the work is not presented in such a way that the markers can easily verify your understanding, it will be difficult to get marks for it. If you need to code your answers to produce results, please paste the completed and annotated source code in the appendix of your submission. Note, that the coursework will be graded on the written report, submitted code is treated as ancillary documentation of your authorship, it not as a substitute for a written answer.
Your coursework submission should not be longer than 7 single sided A4 pages with at least 2 centimetre margins all around and 12pt font. Again, seven pages is a maximum length, shorter courseworks are fine, the code appendix does not count towards page limit, other appendices are not allowed. Coursework over the page limit may incur a penalty. We expect not more than 4 figures including explanatory captions per page.
You are encouraged to discuss the general coursework questions with other students, but your answers should be yours, i.e., written by you, in your own words, showing your own understanding. You should produce your own code to compute your specific answers. You are welcome to reuse code you developed in the lab assignments and interactive computer labs. If you have questions about the coursework please make use of the labs or Piazza, but note that GTAs cannot provide you with answers that directly solve the coursework.

Marks are shown next to each question. Note, that the marks are only indicative.

All coursework is to be submitted on CATE by the specified end date.Please ensure that you are familiar with the straightforward CATE submission process well before the coursework deadline. Unfortunately, emailed or printed submissions cannot be accepted.

Question 1: Understanding of MDPs (20 points)

This question is personalised by your College ID (CID) number.

Consider the following observed traces for a Markov Decision Process with four states $\mathbb{S} = \{s_0, s_1, s_2, s_3\}$, $\gamma = 1$, and immediate rewards specified after every state. Here we only have one possible action ("no choice"), so it is chosen at every step. In other words, the state transitions only rely on transition dynamics. Thus, there is no need for the action to be displayed in the traces. The traces are composed of state $s(t)$ at time step t and immediate rewards $r(t)$ that are collected here upon departing from state $s(t)$. We are going to "observe" a state-reward trace computed from your CID as follows:

- Take your CID number and omit any leading 0s. Then we call the left-most digit $CID(1)$, the subsequent digit $CID(2)$ etc.
- Going from the left most to the right most digit of your CID compute a sequence of states by moving from $t = 1, 2, \ldots$

  $$s(t) = (\text{CID}(t) + 2) \mod 4 \tag{1}$$

  and the sequence of corresponding rewards is

  $$r(t) = \text{CID}(t) \mod 4, \tag{2}$$

  where *mod* denotes modulo operation[1].

Should you CID number without 0s be just 6 digits long (because you have been here for many years, like Aldo), then please prepend a 1 to your CID.

For example, if your CID is 012345678, then the trace of states and rewards is

$$\tau = s_3\ 1\ s_0\ 2\ s_1\ 3\ s_2\ 0\ s_3\ 1\ s_0\ 2\ s_1\ 3\ s_2\ 0$$

We can omit considering the last reward (0 in the example above) as the trace finishes on the last state s_2, but we do not presume whether the last state is a terminal state or a transient state. The trace is just the data that we have observed, we do not know more.

a. Write out your personalised trace. Please use exactly the same format as in the example above. (1 pts)

b. Draw a minimal MDP graph consistent with the data (do not add anything that is not in the data). Please make sure to draw any self-connections between states (these are typically omitted, but it will be beneficial for your learning experience to draw these in). Briefly explain your rationale for the graph you drew. (6 pts)

c. Let us assume we do not know the transition matrix nor the reward matrix of this MDP, just as is the case of a naive reinforcement learning agent. (13 pts)

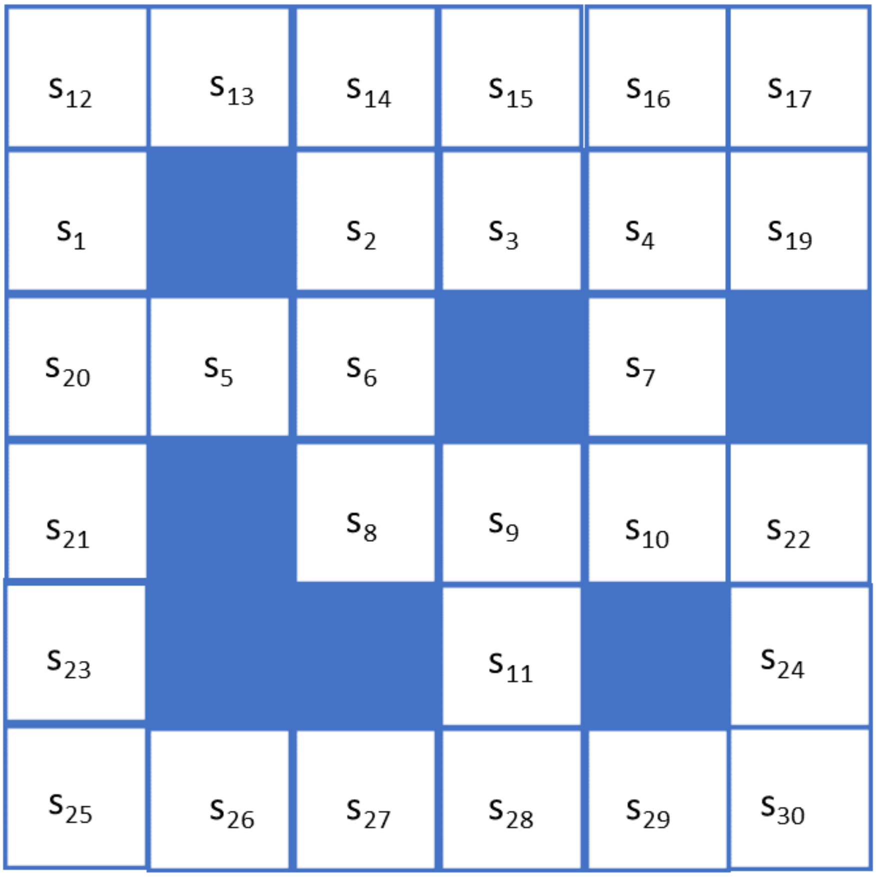
   1. Give an expression of the transition matrix you can infer from the data or the graph you have drawn.
   2. What can you infer about the structure of the reward function and the reward matrix? Briefly give an explanation.
   3. What can you compute about the value of the first state of your trace $s(t = 0)$? Name any method you used and justify your choice.

Question 2: Understanding of Grid Worlds (80 points)

This question is personalised by your College ID (CID) number, specifically the last 3 digits (which we call x, y, z).

Consider the following grid world in the Figure. There are 29 states (be careful there is no state 18), corresponding to locations on a grid. This Grid World has two terminal states. The first terminal state is the reward state as reaching it yields +10 reward and ends the episode. The reward state is the state $s_j, j = ((z + 1) \mod 3) + 1$, where z is the last digit of your CID. The second terminal state is the

[1] https://en.wikipedia.org/wiki/Modulo_operation


s_{12} s_{13} s_{14} s_{15} s_{16} s_{17}
s_1 s_2 s_3 s_4 s_{19}
s_{20} s_5 s_6 s_7
s_{21} s_8 s_9 s_{10} s_{22}
s_{23} s_{11} s_{24}
s_{25} s_{26} s_{27} s_{28} s_{29} s_{30}

penalty state s_{11}, which yields -100 reward. The starting state for simulations of the environment is chosen randomly in each episode. In particular, for each episode the starting state should be chosen from any non-terminal states with equal probability $\frac{1}{27}$ (i.e. there are 27 non-terminal states).

Possible actions in this grid world are N, E, S and W (North, East, South, West), which correspond to moving in the four cardinal directions of the compass. The effects of actions are not deterministic and only succeed in moving in the desired direction with probability p, in which case the action leads to the remaining 3 cardinal directions with equal probability. After the movement direction is determined, and if a wall blocks the agent's path, then the agent will stay where it is, otherwise, it will move. The agent receives a reward of -1 for every transition (i.e. a movement cost), except those movements ending in the terminal state (where you collect the reward for arriving at them).

Throughout this question we set $p = 0.25 + 0.5 \times \frac{(x+1)}{10}$ and $\gamma = 0.2 + 0.5 \times \frac{y}{10}$, where x is the antepenultimate digit of your CID and y is the penultimate digit of your CID.

a. State your personalised reward state, p, and γ (1 pts)

b. Dynamic Programming (with full world knowledge) (24 pts)

   1. Compute the optimal value function and the optimal policy using Dynamic Programming. Briefly state how you solved the problem, including any parameters that you set or assumptions you made.
   2. Report your optimal value function by "writing in" the values into the grid world (handwritten scan is ok).
   3. Report your optimal policy function by "drawing in" arrows reflecting your optimal action for each grid world state (again, hand-drawn scan is ok).

4. Briefly discuss how the value of your γ and p have influenced the optimal value function and optimal policy in your personal Grid World. In particular, you may investigate the effect of having a value of $p < 0.25$, $p = 0.25$ or $p > 0.25$, and similarly $\gamma < 0.5$ or $\gamma > 0.5$.

Note: For those that want to do it by hand and scan: a grid world is reproduced on the last page, you can print this out, write on it and photograph. If you do not use computer typeset answers, please write and draw clearly.

c. Monte Carlo RL (15 pts)

1. Estimate the optimal value function using Monte Carlo (MC) reinforcement learning. Briefly state how you solved the problem, including any parameters that you set or assumptions you made.
2. Report your optimal value function and policy.
3. Plot the learning curve of your agent (reward against number of episodes). In general, a single run will not be sufficient to estimate variability in the learning, thus run your agent a "sufficient" number of times using the same number of episodes. Briefly state what a sufficient number is and how you chose it, and then plot the mean plus/minus the standard deviation of the reward.
4. How does varying the exploration parameter ϵ and the learning rate α of your algorithm impact your learning curves? Briefly explain what you find and discuss and relate it where possible to the theory you learned.

d. Temporal Difference RL (15 pts)

1. Estimate the optimal value function using Temporal Difference (TD) reinforcement learning. Briefly state how you solved the problem, including any parameters that you set or assumptions you made.
2. Report your optimal value function and policy.
3. Plot the learning curve of your agent (reward against number of episodes). In general, a single run will not be sufficient to estimate variability in the learning, thus run your agent a "sufficient" number of times using the same number of episodes. Briefly state what a sufficient number is and how you chose it, and then plot the mean plus/minus the standard deviation of the reward.
4. How does varying the exploration parameter ϵ and the learning rate α of your algorithm impact your learning curves? Briefly explain what you find and discuss and relate it where possible to the theory you learned.

e. Comparison of learners (20 pts)

1. Compare the optimal value function estimation error (the root mean square error between the optimal value function vector computed through Dynamic Programming with your current estimate from reinforcement learning). On an episode by episode basis, plot estimation error against episodes. Compare these estimation error plots between MC and TD implementations. What differences do you see? In your answer relate your reasoning also to the theory of the two approaches that you learned.
2. Train both your MC and TD learners. On an episode by episode basis, plot the value function estimation error of the episode against the reward for that episode. Explain what this plot characterises. Based on this plot also explain how important it is to have a good value function estimate to obtain a good reward. What conclusions can you draw? In your answer relate your reasoning also to the theory of the two approaches that you learned.

3. Is MC systematically better than TD or vice versa at learning the policy quickly and correctly? Does changing the learning parameters or other factors in the algorithms alter this performance relationships. Discuss and provide any evidence you simulated.

Change Log
Version 1.01 Improvements marked in blue.

- We fixed two typos.
- We expanded sentences to make the wording clearer in response to Piazza feedback.
- We provided detail on the modulus function (*mod*).

Name:
CID:
reward state:
$p =$
$\gamma =$

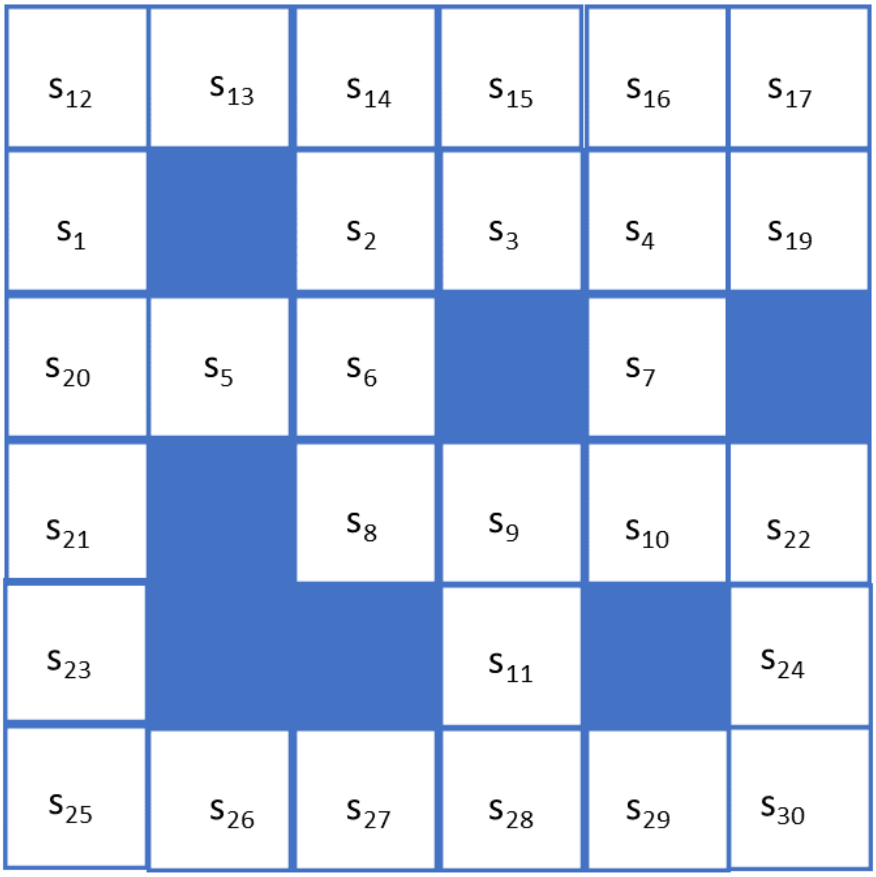


Figure 1: Optimal value function. Values for each state rounded to 1 decimal place. Arrows indicate optimal action direction for each state (deterministic policy), multiple arrows from one state indicate equiprobable choice between indicated directions (stochastic policy).