UNIVERSITY OF MANCHESTER
SCHOOL OF COMPUTER SCIENCE

AI and Games

Time: 14:00 – 16:00

Marker’s feedback version

Please answer any THREE Questions from the FOUR questions provided.
Use a SEPARATE answerbook for each question.
Marks will be awarded for reasoning and method as well as being correct.

The use of electronic calculators is not permitted.
1.

2.

3. a) What is reinforcement learning? How does reinforcement learning differ from supervised learning and unsupervised learning as studied in COMP24111? (4 marks)

Marker’s feedback
Most people got this, as I hoped. However, some people reiterated back at me things about reinforcement learning without saying what it is. Some people were confused about what supervised learning is, saying it is “learning from labelled examples” where the labels tell you if the output is right or wrong. No, that is reinforcement learning. The labels must tell you what the correct output is for it to be supervised learning. I did not deduct marks for incorrect or missing answers about unsupervised learning, because that was not sufficiently covered in the course.

b) Why is reinforcement learning relevant to learning in games? Give another (non-game) situation in which reinforcement learning can be used. (2 marks)

Marker’s feedback
Folks generally got this.

c) In reinforcement learning, often one wants to find a solution which optimises some criterion.

i) One possible criterion to optimise is the “regret”. Give a definition of regret, both in words and mathematically. (3 marks)

Marker’s feedback
Again, most people got this. Some people did not tell me what strategy the regret was measured against. In the mathematics, you need to be sure to define the terms in any equation you write down, which some people failed to do.

ii) Give another criterion which one might want to optimise. (1 mark)
d) Recall the prisoner’s dilemma, with a payoff as follows:

<table>
<thead>
<tr>
<th></th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>C</strong></td>
<td>-2, -2</td>
<td>-10, 0</td>
</tr>
<tr>
<td><strong>D</strong></td>
<td>0, -10</td>
<td>-8, -8</td>
</tr>
</tbody>
</table>

You are going to make a learning bot to play a repeated version of this game, repeated for 200 games, against a community of other bots. You don’t know how the other bots work (and you cannot reverse engineer them) but you can download them and play your bot against them.

i) If you could compute the Nash equilibrium, should you play it? Why or why not? (2 marks)

**Marker’s feedback**
This is a question about why do learning at all. Because, there may be sub-optimal players in the community and you could gain an advantage by learning this. This was not meant to be a question specifically about the prisoner’s dilemma.

ii) Can min-max search with alpha-beta pruning be used to solve this? Why or why not? (2 marks)

**Marker’s feedback**
This turned out to be a hard question. The game is not zero-sum and not perfect information, but it was rare for folks to recognise this. What to do in this case was discussed as part of the mini-max lectures. I did give partial credit for other answers.

iii) Describe, based on methods learnt in this course, how you would make a learning bot to learn to play well against this community of bots. Be sure to describe how you would represent the bot’s strategies, how new policies would be explored, and how strategies would be updated during learning based on the outcomes of the game. In order to make a strong player, the strategies may depend on the history of the opponent in the previous play or plays (such as tit-for-tat, which depends on the play of the opponent in the previous play). (6 marks)
Marker’s feedback

I accepted any learning approach. However, some people did not have learning in their answer. They just said they would try lots of strategies and see which was best. You need some way of generating new strategies.

The most common weakness in many answers was in the representation of strategy; no explicit discussion of the strategy space was given. Aspects of learning were mentioned, such as bandit algorithms for making decisions, and exploration strategies, but the representation of the strategy space was often ignored. These answers were therefore incomplete. As these are repeated games, the best approach would have strategies which depended on the previous actions taken, and discussions of this and how far back to look was desirable. However, credit was also given for cases where history was ignored and strategies for the one-shot game was learned.