CS331: Answers to HW1

Prasad Tadepalli

April 2023

1 Question 1: ChatGPT

The answers to these questions can vary depending on how thoroughly you experimented with ChatGPT. What we are looking for are justified opinions based on examples of good and bad answers.

We have already seen some examples of bad reasoning by ChatGPT in the class. Sometimes it performs impressively, clearly exceeding average human intelligence. At other times it makes mistakes, some almost no human makes.

For example, it was able to solve the 8-queens problem perfectly. Then I asked "can you do the same on a 4 X 4 board"?

It said "no it is not possible," and then went on to give a correct solution, but insisted that it was not correct!

I asked if John is taller than Mary and Mary is taller than Lisa, who is taller, John or Lisa. It said there isn't enough information.

I would say that overall, chatGPT is not always self-consistent. It excels in certain tasks such as composing stereotypical letters, stories and essays. It sometimes answers questions perfectly, but at other times fails on easier questions.

It is conceivable that this kind of approach to Turing Test will eventually succeed, but probably not in a couple of years and not with expert judges. My main reason is that chatGPT and similar systems are based on statistical pattern recognition, and can be fooled by asking clever questions that are trivial to humans but are not directly reflected in the statistics of text.

For example, I asked which is larger, a square of side R or an enclosing shape. It said there isn't enough information to tell. It was able to answer the question correctly by calculating the areas when the phrase "enclosing shape" is replaced with "enclosing circle". Average humans can answer the questions easily even if they are not able to derive the areas of squares and circles.

2 Question 2: Agents

As many of you have noticed, a simple reflex agent can only clean part of the room. Here is an example agent.

```
If dirt then Succ
Else MoveRight
```

Since the simple reflex agent cannot detect the end of the corridor, it keeps trying to move right. If it is originally in position x and there are N cells, all positions to the left of x cost it 5 units each time step, which gives a cost of 5(x-1)t. The cost of other positions start at (N-x+1)*5 and goes down to 0 in 2(N-x+1) steps. Finally the action cost will be t in t steps. Adding it all up, we

get 5(x-1)t+5*2(N-x+1)(N-x+1)-5(N-x+1)+t. The important part here are the first and the last terms 5(x-1)t+t which increase linearly with time t.

Model-based reflex agents can do better because they can remember the previous actions and turn back when needed. I use a bit 'back' to indicate when the agent is going back after hitting the wall. The bit 'pass' indicates that the agent is going back and has not yet seen any dirt. After it sees some dirt it starts cleaning and resets 'pass'. In that mode if it sees a clean cell after executing a non-suck action it stops. Otherwise it keeps going left and cleaning dirty cells. I use the third bit 'sucked' to remember that the agent executed 'suck' in the previous step.

```
if back and dirt suck, set(sucked),reset(pass);
elseif back and sucked MoveLeft, reset(sucked);
elseif back and not(pass) halt;
elseif back MoveLeft;
elseif dirt suck, set(sucked);
elseif sucked moveRight, reset(sucked)
else moveLeft, set(back),set(pass)
```

The agent is going to clean all the cells and incurs no cost for permanent dirty cells. It takes (N-x+1)2 steps to reach the right end (one Suck action and one MoveRight action for each cell on its right). It takes N MoveLeft actions and x-1 suck actions when going left. The dirty cells on the right cost 5*2*(N-x+1)(N-x+1)-5(N-x+1) similar to the previous calclulation. The dirty cells on the left cost 5*2*(x-1)*(x-1)-5(x-1)+5*(x-1)*(3*(N-x+1)). Hence the total cost is (n-x+1)2+N+x-1+10(N-x+1)(N-x+1)-5(N-x+1)+10(x-1)(x-1)-5(x-1)+15(x-1)(N-x+1).

One corner case that is worth noting here is when the agent starts in the leftmost cell. In this case it could stop after the first pass, but it would not have known that this was the case, it returns to its leftmost cell and keeps trying to go left. Hence it pays an additional 1 unit of cost each time step while trying to go left.

3 Environments

	Go	**SuDoKu**	**Internet Shopping**	**Jigsaw Puzzles**	**Scheduling**
Observable	Fully	Fully	Partially	Fully	Partially
Agent	Multi	Single	Single	Single	Single
Deterministic	Yes	Yes	No	Yes	No
Episodic	No	No	No	No	No
Static	Yes	Yes	No	Yes	No
Discrete	Yes	Yes	Yes	No	Yes
Known	Yes	Yes	No	No	Yes

Most of these are straightforward. Internet shopping is partially observable because one is only able to access a small number of web pages at a given time. Most of the action is in determining which site to go to. This is still a "single agent" domain because you don't typically think about other buyers and sellers when shopping unless you are in an auction setting. It is not deterministic or static because the websites keep changing especially with products like airline tickets. There are usually many actions to be taken in sequence, and agents typically do not have a good model of what actions do even at the level of result distributions.

I said meeting scheduling is partially observable because one need to collect information about the availabilities by polling. One could begin by polling all or poll one person and only then poll others based on when they are available. Which is better depends on who is typically more busy,

etc. So there is advantage to treat it as partially observable domain. It is not multi-agent though, because the other people can be viewed as stochastically responding to queries. It is also not static. If you take too long in scheduling, the calendars might change for other people. It is debatable whether it is known or not known depending on the people you are scheduling. Sometimes you might know their typical schedules (how busy they are) and sometimes you may not.

I said Jigsaw puzzles are continuous because they need to be rotated by some real-valued degrees.