4705 HW 1

9/11/23

(1)

Intelligence - The ability to reason and make rational decisions.

Artificial Intelligence - The ability to pursue a goal in the face of obstacles.

Agent - An entity that can take in information about it's environment and use that data to behave intelligently

Rationality - The ability of an agent to take in information then utilize it to behave in a way that achieves a goal optimally while there are other potential behaviors the agent can exhibit.

Logical Reasoning - The ability of an agent to take in information then adhere to a sequence of guidelines in order to obtain an optimal solution.

Agent Function - Like an algorithm, takes some inputs which are fed into a defined sequence of instructions and depending on what the input data was, yields different results.

Agent Program - The actual implementation of the Agent function in order to yield real world data.

Autonomy - The ability of an Agent to operate without manual external factors.

(2) Reflex actions such as flinching from a hot stove are rational because they are essentially programmed into organisms in order to achieve a certain goal which in this case is to avoid being burned. Reflex actions also achieve this goal optimally since by definition a reflex occurs without concious thought yet as soon as a reflex senses the presence of danger it immediately acts to avoid the danger.

Reflex actions however are not intelligent because there is no reasoning happening, instead as soon as an agent takes in input the reflex yields an output as a result of programming from evolution.

(3) Although the vast majority of people have no idea how to convolve an image with a Gaussian, through the process of abstraction which is in esscence what all of computer science is, it is perfectly sensible to say that the vision system is doing the math. Extrapolating from the example of computer science, modern CPU's are capable of doing billions of arithmetic computations every second which an individual is obviously incapable of doing but by writing an algorithm is able to abstract away all the difficulty of doing such tasks themselves.

(4) Given:
- one point gained for each clean square per time step.
- 1000 time steps
- only possible actions Left, Right, & Suck.
- Agent knows what it is & if there is dirt there.

(A) The function defined in class: is indeed rational because it avoids redundant checks. for instance the table lookup algorithm we also disussed would accomplish the same task however by keeping a table of the percept sequence our space complexity becomes unfeasible. The rational function we disussed avoids such unnecessary computations and utilizes simple conditionals to decide what do next and requires only constant space.

```
function Clean() {
  if Status = dirty:
    return SUCK.
  elif location = A:
    return right
  elif location = B:
    return left.
```

(b) Rational agent function if each movement costs 1 point:

```
function Clean(Prev_location)
if Status = dirty:
    Prev_location = location
    return Clean
elif (location = A & Prev_location = dirty)
    Prev_location = location
    return right
elif (location = B & Prev_location = dirty)
    Prev_location = location
    return left.
return Stay.
```

could just Put it here once instead.

If a rational agent is taxed 1 point everytime it moves then in order for it to be rational it should only move when necessary. The code above describes how a rational agent should behave if it costs points to move. It does require an internal state that tells it where it was last, which in this case can only be A or B, so that it can figure out whether or not it needs to clean another room. One of the assumptions in this problem is that there is no NoOP action however a rational agent in this context should stay where it is if it doesn't need to clean anything

(c) If we have an unknown environment where previously cleaned tiles can become dirty again:

- Simple Reflex agent wouldn't work since we need previous percepts such as recently cleaned tiles

Model based reflex agent might work since the agent can't see the current state of previous rooms however there would need to be a mechanism so the agent can learn how the world evolves so it's aware of which rooms to revisit in the future.

Utility & Goal based agents would be difficult to utilize for such a scenario

A learning agent that learns through experience would be the most applicable because after going through enough trials it is capable of learning how often & which tiles get dirty & it can utilize that information to figue out which rooms to visit in the future.

⑤ PEAS: Performance Measure, Environment, Actuators, Sensors

a) Playing soccer:

P: #goals, # assists, player ratings

E: field quality, goal size,

A: Player legs

S: Player Eyes

fully observable,
Dynamic,
discrete,
Sequential, (depending on score ex. agent might choose to play more forward or further back)
Stochastic,
Multi Agent

b) Exploring subsurface of oceans on a planet:

P: amount of data collected, quality of data

E: depth of water, animals & other life in water.

A: propellers to traverse water. Communication equipment to transmit data

S: Ultrasonic sensors, proximity sensors,

Partially Observable, Dynamic (animals could be moving around), Continuous (constantly changing environment & agent can choose to observe anything leading to endless possibilities), Episodic (previous percepts doesn't affect current decision), Stochastic (ex If agent aims to go somewhere it might end up in a different place due to ocean currents), Single agent.

c) Shopping for used AI books:

P: Price, shipping time

E: Website, Seller marketplace

A: Mouse / Keyboard, Payment method

S: eyes to view monitor.

fully observable (theoretically you could search through every book on each website.

dynamic
discrete
Single agent

Stochastic (keep track of previous books seen)
Sequential

(someone could buy the book while you are deliberating).

d) Tennis match:

P: Score, Shot accuracy

E: Ball, Tennis court, net, racquet

A: Legs & Arms

S: Eyes

fully observable
Static
discrete
Sequential
Stochastic
multi agent

e) Practicing tennis against a wall:

P: consistency

E: Ball, wall, racquet

A: Legs & Arms

S: Eyes

fully observable
static
discrete
Episodic
stochastic → (Ball might not go exactly where you want it to go every time)
Single Agent

(f) Performing High Jump:

P: height of jump, consistency

E: mat, bar

A: Legs & Arms

S: Eyes

fully observable
Static
discrete
Episodic
deterministic → (depending on how much force you used to jump you should attain the same amt of height)
Single Agent

g) Knitting a Sweater

P: time taken to knit, quality of knit

E: sow, yarn

A: hands

S: eyes

fully observable
Static
Continuous → (current knit depends on last knit)
Episodic
deterministic
Single Agent

h) bidding on an item:

P: Amount of acquired items, cash remaining
E: Auction house, bidders, item
A: hands, mouth to bid
S: eyes, ears to see/hear bids/items for sale.

Partially observable (don't know what other bidders want or funds.
Dynamic (someone could bid while you are making a decision.
Discrete
Seauential (winning/losing Past bids might influence current bid.
Stochastic (other People bidding can't be fully predicted.)
multi agent.

Fully Observable: Can Agent see the full environment or only Partial?

Static vs Dynamic: Does the environment change while Agent makes a decision.

Discrete vs Continuous: Are there a finite number of inputs/outputs or limitless possibities.

Sequential vs Episodic: will Past actions affect current actions or is each action Self-Contained.

Deterministic vs Stochastic: Can the state of the environment after the action be predicted or is there randomness involved.

(6) Reflex agent - No regard for previous State of system, current action only depends on current percepts.

Model-Based Agent - Maintains a model of how environment was previously to help make current decision.

Goal-Based Agent - Along with percepts, has a goal to influence actions so it will take into account whether the action will get it closer to acheiving the goal.

Utility-Based Agent - Along with percepts, contains a utility function which it seeks to maximize in it's decision making.

Learning-Agent - Contains A Learning & Performance function which work together to tell the agent how well current action did & how to improve.

(7) a) Murphy's Law: Vaccum fails to suck dirt 25% of the time & Sensor is wrong 10% of the time.

In order to reduce error the vaccum could be programed to Sense → Sense → Suck → Suck. By doing so the sensor would be incorrect only 1% of the time & fail to suck only 6% of the time however it would now take twice as long to clean.

b) Small Children: After every time step each clean square has a 10% Chance of becoming dirty again.

For such a scenario it would be ideal to have a model or learning based agent that is able to keep track of where children play & which areas frequenty get dirty. Since any tile has a 1/10 chance of getting dirty it would need to frequently revist areas it recently visited.