COMS W4701: Artificial Intelligence
Written Homework 1

Huibo Zhao (hz2480)

September 27, 2017

Problem 1

This essay is written by Turing in 1950 and people believe it is the first comprehensive paper that analyzes "artificial intelligence", which was a new concept to public at that time. Also, Turing introduced his famous Turing Test theory in this paper. He generally listed nine objections on why people believe human cannot build "machines" that can think like human. I think Turing really did some great refutations on some claims. For example, people criticized that machines can only do what people programmed it to do and they cannot learn from experience. Now, as machine learning becomes one of the most popular topic of computer science and AlphaGo has proved to have the ability to beat human by learning with numerous past experiences. Though today's artificial intelligence still cannot surprise us that much, it at least proves its ability of learning from past. However, one objection that I think still carry weight now is that machine cannot have people's feelings. Though AlphaGo beat best human player, AlphaGo itself cannot feel happy or excited about this. Therefore I don't think we can state that machine can think like human unless machines can also have emotional feelings as us.

One objection I can think is that it is still very hard for letting the machine learning from outside environments and develop their own thoughts. For instance, some people may feel warm if a stranger hugged him/her but some may feel uncomfortable. Nowadays technology has no problem recognizing the "hug" behavior but how can it make machine generate actual feelings about this behavior? Another interesting objection that I saw someone mentioned is that ai does not have the instinct of being afraid of death. People were given this instinct when born but machine can never be achieved real artificial intelligence until it can develop the instinct of fearing death.

Problem 2

(a) Rock-Paper-Scissors AI
performance measure: maximize winnings
environment: rules of the game, previous move, opponent's move
actuators: toss out rock/paper/scissor hand gesture
sensors: perceive opponent's hand gesture

fully observable: the agent can fully perceive opponent's current move (hand gesture) multi-agent : competitive multi-agent systems. The opponent can be another ai

deterministic: environment states are completely defined by agent's actions

episodic/sequential: it can be both episodic and sequential though I personally prefer episodic because the probability of winning does not depend on past experience thinking in math. However, there may exist a special order that ai can follow to increase the winning probability by analyzing past moves

static: the environment will not change until the agent takes an action

discrete: thinking and acting can be discrete states in this case

is learning required?: This category has similar issue as episodic/sequential. If considering the winning probability does not depend on past moves, then learning is not required. If considering the winning probability depends on past moves, then learning is required.

(b) Assembly line robot installing wheels on a car
performance measure: how well the installed wheels fit the car
environment: given wheels, cars and orders
actuators: install the wheels, proceed working by handing in installed car and accept next car
sensors: perceive the size, location of the wheels and cars

fully observable: the robot can fully perceive the installing environment and parts

multi-agent: collaborative multi-agent systems. They robots may work together

deterministic: environment states are completely defined by how robots install the car

sequential: previous installing actions definitely have effects on next actions

dynamic: the environment may change (installed wheels fall off or some other accidents)

discrete/continuous: It can be both. In general, if you state that the robot can think for a while then do the installation, then it is discrete. However, if you state that during the installation, the robot must perceive and act simultaneously and continuously, then it is continuous

is learning required?: Yes/No. If the installing parts are all at the same size, the robot may only need to follow a standard installation procedures without altering it. However, the robot may also grow installation experiences by installing more and more cars

(c) Customer service chat bot
performance measure: how well it can communicate with the customers

environment: the service that chatbot can provide, what customers say
actuators: respond the customers
sensors: perceive customers' words and understand what customer wants for answers

fully observable: the chat bot can fully perceive what customers send

single agent: the chat bot only communicates with human, not another chat bot

stochastic: the chat bot does not what customers would ask or respond

episodic/sequential: It can be both episodic and sequential based on if customer is continuously asking about one same issue or ask different question each turn

dynamic: when the chat bot is thinking what to respond, the environment may change. For instance, the customer may ask another question or make new requests

discrete: the percepts and actions are not continuous

is learning required?: Yes. For a chat bot, it may grow its capability of talking to customers by analyzing the feedbacks

(d) An email spam filter system
performance measure: how well it can filter those spam emails
environment: the details about the both normal emails and spam emails
actuators: mark certain emails as spam and filter them
sensors: continuously receive upcoming emails

fully observable: the filter system can fully access the email contents

single agent: the filter system is a single agent that deals with classifying emails

deterministic: whether the emails are marked as spam or not are completely determined by filter system

episodic: this is similar to image classification. It is episodic.

static: the environment will not change as the filter system classifies the emails

discrete: the percepts and actions are not continuous

is learning required?: Generally no if the system already contains enough knowledge on distinguishing spam emails. However, if the system needs to learn as it processes, then the answer will be yes

(e) Autonomous rescue robot that locates and evacuates victims after natural disasters performance measure: how accurate it can predict the location of victims and how well it actually evacuates the detected victims
environment: the disaster environment, the victims, other rescue robots
actuators: detect the locations of victims and report them, evacuate victims with various methods depending on different types of disasters
sensors: cameras, radar systems, GPS, various sensors

partially observable: the robot cannot fully access the whole environment

multi-agent: collaborative multi-agent systems. The robots can work together to better rescue people
stochastic: this is similar to driving a car. Environment states are not completely defined by actions

sequential: It is sequential because evacuate victims is a series of events

dynamic: we all know natural disasters are very dynamic

discrete/continuous: can be both. For example, after detecting the location of victims, the robot may take a moment to think what next action should be, in this case it is discrete. During the process of actual evacuating victims, percepts and actions may be required as continuous, in this case it is continuous.

is learning required?: Yes. The robots need to continuously learn from the outside environment to better help people

Problem 3

A water jug puzzle: You need exactly 4 gallons of water. You are near a spring and you are given two jugs, a 5-gallon one and a 3-gallon one. You can fill either jug from the spring and you can pour water between the jugs. Neither jug has any measuring markings on it, so when you pour water into a jug you must fill it completely to get a precise measurement. You can also empty a jug on the ground. How can you get exactly 4 gallons of water?

1) Model this problem formally as a search problem. Describe what each state looks like, define an initial state and describe the goal test. What are the possible actions and in what situation is each action applicable? Formally describe the result state of each action.

Each state: how much water are in the two jugs. For instance, (0,5),(0,3) means that both two jugs are empty. (5,5),(0,3) means that the 5-gallon jug is full and the 3-gallon jug is empty

Initial state: (0,5),(0,3) it starts with both jugs being empty

Goal test: when the sum of water of two jugs is exactly 4 gallons

Possible actions: Get water from spring and fill up to full. Transfer water from one jug to another jug that is not full. Pour water to the ground. The rules for transferring are we would either pour the water until the accepting jug is full or until the pouring jug is empty. We could not choose the exact amount of water for transferring because the jug does not have any measuring scales on them.

If we get water from spring, then the accepting jug will be filled up to full. If we pour water to the ground, then the pouring jug will be empty. If we choose to transfer, we need to consider the above transferring conditions. Generally, the accepting jug will gain x gallons water and the pouring jug will lose x gallons water.

2) Provide a path in the state space that solves the problem (as a sequence of states connected by a sequences of actions). To find this solution, it helps to draw out the (part of) the state space as a graph.

We start with initial state (0,5),(0,3).
$->$ Get water from spring to fill up the 3$-$gallons of water: (0,5),(3,3)
$->$ Transfer water from 3$-$gallon jug to 5$-$gallon jug: (3,5),(0,3)
$->$ Get water from spring to fill up the 3$-$gallons of water: (3,5),(3,3)
$->$ Transfer water from 3$-$gallon jug to 5$-$gallon jug: (5,5),(1,3)
$->$ Pour the water in 5$-$gallon jug to the ground: (0,5),(1,3)
$->$ Transfer water from 3$-$gallon jug to 5$-$gallon jug: (1,5),(0,3)
$->$ Get water from spring to fill up the 3$-$gallons of water: (1,5),(3,3)
Now, we reach the goal test where the sum of water of these two jugs is exactly 4 gallons

Problem 4

Given a function being admissible, we know that this function $h(n)$ <= h*(n), where h*(n) is the true cost to the goal.

Since we want a new heuristic function that provides an estimate of the cost to the goal that is as close as possible to the true cost, I would obtain it by taking the heuristic function that gives the max value. In other word, for each state s, $h(s)$=max($h1(s), h2(s), ..., hn(s)$)

It is easy to prove this new heuristic function is still admissible: Since every heuristic function in this set is admissible, then taking the heuristic function that gives the max value from this set must still be admissible

Problem 5

Is the bunny guaranteed to find the carrot? Does the search strategy guarantee that the path with the shortest number of moves will be found?

a) Depth First Search: It is guaranteed to find the carrot if we keep track of the nodes that we already searched so that we do not stuck in an infinite loop. It is not guaranteed to find the carrot if we do not keep track of the nodes. Secondly, it does not guarantee that the path is optimal because dfs searches the recently found node first therefore does not guarantee the shortest path.

b) Greedy Best-First Search using the Manhattan Distance between the bunny and the carrot as a heuristic: It is guaranteed to find the carrot if we keep track of the nodes that we already searched so that we do not stuck in an infinite loop. It is not guaranteed to find the carrot if we do not keep track of the nodes. (Suppose the bunny is stuck by the obstacle, the bunny has to move around the obstacle to reach the goal. However, if we do not keep track of the nodes, when the bunny starts to move around, the system will move the bunny back to the original position because it has the shortest manhattan distance, though being stucked). Secondly, the strategy does not guarantee the path is optimal. As just mentioned, the existence of the obstacles can stuck bunny and force the bunny to move around, so the path may not be the shortest.

c) A* Search using the Manhattan Distance between the bunny and the carrot as a heuristic: It is guaranteed to find the path and being optimal because a* algorithm will keep updating the nodes for finding the optimal path.

d) A* Search using the Straight-Line Distance between the bunny and the carrot as a heuristic: It is guaranteed to find the path and being optimal. The reason is the same as above. It is due to the properties of a* algorithm. As long as the heuristic function makes sense, we should be able to find the optimal path.

If the bunny is allowed to hop to a diagonal grid cell, then answer for b,c,d will not change. It is because this condition only gives more successors on each node, it does not have effect on searching algorithm.