CS 4341: Homework 1

Adam Camilli (aocamilli@wpi.edu)

January 25, 2018

Ch 1: Introduction

1. **Problem 1.7:** To what extent are the following computer systems instances of artificial intelligence?

   - Supermarket bar code scanners.

     Upon sensing a barcode provided by a buyer or cashier they simply look up the barcode in a linked database. They are a good example of electrical engineering backed up by a very elementary DB-lookup program. The process of reading a barcode is not akin whatsoever to human sight: The bars are simply a way of representing numbers that can be read by a photoelectric cell that converts the zebra pattern into on-off pulses representing binary numbers. Nothing about this process is autonomous, let alone intelligent. Therefore, supermarket bar code scanners are not a good example of AI.

   - Web search engines.

     Web search engines have a number of features that can be interpreted as intelligent, such as use of data mining to detect typing errors. Every modern search engine makes use of machine learning and natural language processing as well. However, it is hard to argue that search engines themselves are rational agents: Their ability to "learn" is extremely specialized and not at all extendable to other classes of problems. Search engine design is mainly based in development of algorithms such as PageRank in combination with efficient DB design and processing power. Search engines do utilize many techniques from AI development and so are a much better example of AI than bar code scanners. But they are not autonomous and are only intelligent in certain very specific instances.

   - Voice-activated telephone menus.

     Obviously these devices are a fantastic example of the power of natural language processing, considered one of the hardest features to implement in AI. But these devices are not really intelligent whatsoever: Like search engines, they are able to analyze certain kinds of patterns in a way that is similar to human, but this ability is rooted in very specialized engineering that is not extendable to general pattern recognition. Some speech recognition systems can become attuned to a certain speaker's voice, but telephone voice systems are generally speaker-independent devices which cannot do so. As experience with Siri shows, however, deep learning has enabled these devices are to interact with humans in a dynamic and improvised manner. This represents a much bigger advancement than barcode scanners which simply rely on lookup tables, and the intelligence required can be argued to act rationally in a manner search engines cannot: Siri needs to string together sentences, a task far harder for computers than displaying a list. Voice-activated systems are therefore an even better example of AI than search engines, though they are definitely not a complete rational agent.

   - Internet routing algorithms that respond dynamically to the state of their network.

     Although these algorithms make little to no use of natural language processing, they are required to act rationally in terms of cost-benefit analysis, to a much greater extent than search engines or voice processors. Routers need to accomplish several goals at once: The goal of being fast, the goal of transmitting complete and non-corrupted data, not consuming an excess amount of power, etc. The algorithms to dynamically work to accomplish these tasks are great examples of how a device acts rationally. With constantly varying input (network status), they are able to adjust how their output (packets) is delivered. Out of all the devices, it can be argued this is the most complete example of a rational agent, and therefore of AI.

2. Do you think that the Turing test is a good way to judge a rational agent? Justify your answer.

   The Turing test is useful as a way to understand the metrics of how a machine can pass as human. But it is far too abstract and difficult to determine rationality, which a machine can easily possess while still being machinelike.

Ch 2: Agents

1. Problem 2.4

Agent Type	Performance Measure	Environment	Actuators	Sensors
Robotic Agent: Soccer player	Win-Loss record, goals for-against, possession	Field, ball, own/other team, own body, teammate and opponent body	Legs for moving, feet/head/body for manipulating ball, arms/appendages for balance/fighting opponents	Camera, touch and orientation sensors, speedometer, accelerometer
Robotic Agent: Titan subsurface ocean explorer	Data retrieved (pictures and video)	Ice desert, water lakes under ice	Legs for traversing surface, drill to break through ice, propellers for moving, rudders+fins for steering	Camera, thermometer, mass spectrometer, chemical sensor (to detect non-water lakes), orientation sensor, speedometer, radio/satellite receiver
Software Agent: Used AI book shopper	Obtaining used books pertaining to AI from online markets	Internet	Follow link, submit queries, enter data in fields	HTML parser, scam detector, algorithm to use website data to determine if products are used and about AI
Robotic Agent: playing tennis against an opponent	Win-Loss record, not making unforced errors	Turf and clay courts, ball, racket, own body, opponent body	Legs for moving, feet for balancing, gripping appendages	Camera, separate ball and opponent spacial trackers, orientation sensor, joint controllers for switching between forehand and backhand
Robotic Agent: playing tennis against a wall	Repeatedly hitting and returning ball	Half-sized court with wall instead of net, own body, ball, racket	Legs, feet, gripping appendage	Camera, ball tracker, orientation sensor, joint controllers
Robotic Agent: Performing a high jump	Jumping over as high a bar as possible without touching it	Straight track, bar, mat	Legs, optimized leg joint controllers to jump as high as possible, additional joint controllers to elongate body during jump and twist body before jump	Camera, orientation sensor, own center of gravity detector, distance tracker to bar
Robotic Agent: Knitting a sweater	Knitting a wearable sweater quickly and efficiently	Pile of yarn on spool, knitting needles, frame to hold fabric and spin yarn	Appendages with digits to manipulate yarn and vary stitches, needle operating device + motor, motor to spin yarn	Camera, pattern recognition + scheduler
Software Agent: Bidding on item at an auction	Outbid others (obtain items) as economically as possible	Item list, auctioneer, own bid, other bids	Bid signaler (hand, data submitter, etc.), exit function when done	Auction start sensor, bids vs real-value tracker, current highest-bid tracker

2. Problem 2.5

Term	My definition
Agent	An autonomous entity that observes its environment with sensors, and then intelligently acts upon the environment with actuators.
Agent Function	A function that mathematically relates an agent's percept sequence to an action
Agent Program	A computer program that takes input from an agent's sensors and outputs an action to its actuators
Rationality	The concept of maximizing the usefulness of one's actions given one's current knowledge
Autonomy	The ability of an agent to act for itself, independent of the influence of any other agents or entities
Reflex Agent	An agent that only acts in response to a direct stimulus, using only its immediate perception of the stimulus to make a decision.
Model-Based Agent	An agent that possesses knowledge of its environment in the form of a mathematical or scientific model, and can use it to reach decisions.
Goal-Based Agent	An agent that possesses knowledge of its environment and of a goal it desires to reach. It collects multiple possible actions using perception and a model, and selects the action that matches its goal.
Utility-Based Agent	An agent with knowledge of the world and its goal, and an internal utility function to quantify how useful an action is in terms of reaching its goal. It then chooses the action that matches its goal and has the highest utility.
Learning Agent	An agent with the ability to internalize what it perceives and become more effective as it gains more knowledge of its environment. It has a learning element to make improvements, and a performance element to select actions. The learning element ideally increasingly optimizes the performance element as knowledge is gained.

Ch 3: Search

1. Problem 3.2

(a)
- States: A graph whose nodes can point north, south, east and west, or to a sentinel node when adjoining a wall; Current orientation; Current node
- Initial state: Graph with one central node, pointing north.
- Actions: Turn north, south, east, west; Move forward
- Transition Model: Move in current direction if not facing wall, if facing wall stop.
- Goal Test: Is most recent node out of the maze?
- Path Cost: None

State space is infinite.

(b)
- States: A graph whose nodes can point north, south, east and west, and which indicate if there is a neighbor in that direction; Current orientation; Current node
- Initial state: Graph with one central node, pointing north.
- Actions: Turn north, south, east, west if at an intersection; Move forward
- Transition Model: Move in current direction if not facing wall or at an intersection, if facing wall stop, else if at an intersection turn clockwise (north$\rightarrow$ east$\rightarrow$ south $\rightarrow$ west)
- Goal Test: Is most recent node out of the maze?
- Path Cost: None

State space is infinite.

(c)
- States: A graph whose nodes can point north, south, east and west, and which indicate if it is a turning point; Current orientation; Current node

- Initial state: Graph with one central node, pointing north.
- Actions: Turn north, south, east, west if at a turning point; Move forward
- Transition Model: Move in current direction if not facing wall or at an intersection, if facing wall stop, else if at an intersection turn clockwise (north$\rightarrow$ east$\rightarrow$ south $\rightarrow$ west)
- Goal Test: Is most recent node out of the maze?
- Path Cost: None

State space is infinite, current orientation is now irrelevant.

(d) i. The passages are all straight and passable at equal costs
ii. The robot has unlimited energy
iii. The maze is oriented perfectly north, south, east and west

2. Problem 3.6

(a) 4-Color

- States: A graph where nodes correspond to regions and edges are drawn between nodes representing adjacent regions; Current node; Node queue.
- Initial state: Planar graph with uncolored nodes.
- Actions: Color node; Add child(ren) to queue
- Transition Model: Add all nodes to queue. Color current node, remove it from queue, move to next node in queue.
- Goal Test: Graph is colored (no more nodes in queue) and no adjacent nodes have same color.
- Path Cost: None
- State space is bounded at 4^n where n is number of regions (nodes).

(b) Monkeys.

- States: Square room with $n * n$ tiles, n^2 possible banana arrangements; Current node (monkey); Has bananas; Has crates; Current height.
- Initial state:
- Actions: Move up, down, left, right; Move crates up, down, left, right; Climb crates; Grab bananas.
- Transition Model: Move up, down, left, right. If nothing found move random direction. If wall found, turn around. If crate found, pick it up. If two crates found, plant and climb. If no bananas found, climb down, pick up crates, move random direction and repeat.
- Goal Test: Monkey has the bananas.
- Path Cost: None
- Uses breadth-first search of room.

(c) Input records.

- States: An array of records, one or more illegal; Current index; List of indices of illegal records.
- Initial state: Array of records, current index 0, empty illegal list.
- Actions: Inspect record, move to next record.
- Transition Model: If illegal, add index to illegal list. Else do nothing. Repeat for whole array.
- Goal Test: Inspect whole array.
- Path Cost: None (illegal or not, assumed negligible)

(d) Jugs.

- States: Three jugs with different states of fullness.
- Initial state: All three jugs empty.
- Actions: Fill desired jug, empty into desired jug, empty desired jug on ground

- Transition Model: Jug can be completely filled, completely emptied, or increased/decreased by size of another jug.
- Goal test: Does any jug contain exactly one gallon?

3. Problem 3.10

Term	My definition
State	A collection of parameters and information representing an agent's perception of its environment at a particular moment in time.
State Space	The set of all possible states in a given system.
Search Tree	A tree data structure used to hold keys of a set so they can be traversed or searched for by an agent in an algorithmic fashion.
Search Node	A key in a search tree connected to a parent key (unless it is the root node) and to 0, 1, or many child keys.
Goal	A state that is desirable to an agent.
Action	A change from one state to another by an agent.
Transition Model	A mathematical or logical model describing the results of a set of actions.
Branching Factor	The average number of children of each node in a search tree.

4. Problem 3.14
   (a) False: Depth-first search may sometimes directly find goal with no backtracking, which outperforms A*.
   (b) True: $h(n)$ never over-estimates remaining optimal distance to goal node.
   (c) False: A* still works in continuous situations as a heuristic.
   (d) True if branching factor is finite. If this is the case any goal node at a finite depth will be found.
   (e) False: Rooks can move many spaces in a straight line, and so Manhattan is not admissible.

5. Problem 3.21
   (a) True: If step costs are equal, $g(n) \propto$ depth.
   (b) True: Depth-first search is Best-first search with $f(n) = depth(n)$.
   (c) True: Uniform-cost search is A* with $h(n) = 0$.

6. In this problem, you are asked to select the best search strategy for each of the situations given below. You can choose any of the informed or uninformed search methods discussed in class. Justify your answer briefly but decisively.
   (a) All the arc (= action) costs are equal. No heuristic information is given. Time is not a problem, and you have plenty of space. But you must find an optimal solution.

   Since we have no heuristic and time is not an issue, we need to select an uninformed search strategy that is complete and finds an optimal solution. Due to its ease of implementation, I suggest bread-first search.
   (b) Cost and heuristic information are given. You need to find an optimal solution and you want to save as much time as possible.

   In this case, choose A*: It is optimal and complete, and most of the time is the most efficient heuristic search.
   (c) c) The search space is large and the depth of the solution is not known. No cost and no heuristic information are given. You need to find the shallowest solution but you don't have much memory/space available.

   Here we need an uninformed search strategy that only goes as deep as it needs to, but is also complete and optimal. Therefore I suggest iterative deepening search.

7. Problem 11 attached separately.
8. Problem 5.9

   (a) Upper bound is $9! = 362880$, true figure (since games end 5, 6, 7, 8 moves) is 255168.
   (b) to e. attached separately.
9. Problem 13 attached separately.