COMS W4701: Artificial Intelligence, Summer 2022

Homework 1 Solutions

Problem 1: HODL (15 points)

(a) **States**: The state space S is defined by the current conditions of the crypto, so $S = \{(r, c, e) | r \in \text{risk levels}, c \in \text{current market conditions}, e \in \text{events}\}$
Actions: Actions may include choosing the type of trade (buy or sell), choosing the amount of crypto to trade, and choosing the time of the trade. This is realized as the set $A = \{(o, n, t) | o \in \{\text{buy}, \text{sell}\}, 0 \leq n \leq \text{currency in circulation}, \text{current time} \leq t \leq \text{whenever the crypto crashes}\}$

(b) **Properties**:
a. Partially observable: Trader agents cannot know every piece of relevant information when making decisions. For example, they can't know what actions another trader will take.

b. Multi-agent: Because prices are subject to demand, the action of other trading agents affects the market conditions.

c. Stochastic: The market conditions are insufficient for predicting future states of the market. The unknown actions of other agents makes it fundamentally unpredictable.

d. Sequential: Sequential because trade activity is cumulative over time

e. Dynamic: Conditions change constantly, so the environment will change as the agent thinks.

f. Nominally discrete because there is only a countable infinity of states and actions, but can also be justified as continuous.

Problem 2: Monty Hall (15 points)

(a)

$$
\begin{aligned}
\mathbb{E}[U] &= 1 \cdot \mathbb{P}(U = 1) + 20 \cdot \mathbb{P}[U = 20] - 100 \cdot \mathbb{P}[U = -100] \\
&= 1 \cdot \frac{48}{50} + 20 \cdot \frac{1}{50} - 100 \cdot \frac{1}{50} \\
&= -\frac{16}{25} = -0.64
\end{aligned}
$$

(b) This depends on the box you've chosen in the first place. If you initially chose a box with a single bonus point ($p = \frac{48}{50}$), then the expected utility is: $1 \cdot \frac{42}{44} + 20 \cdot \frac{1}{44} - 100 \cdot \frac{1}{44} = -\frac{19}{22}$. If you initially chose a box with the free A on the final ($p = \frac{1}{50}$), then the expected utility is: $1 \cdot \frac{43}{44} - 100 \cdot \frac{1}{44} = -\frac{57}{44}$. If you initially chose a box that makes you retake the class ($p = \frac{1}{50}$), then the expected utility is: $1 \cdot \frac{43}{44} + 20 \cdot \frac{1}{44} = \frac{63}{44}$ The expected utility all together is thus:

$$\begin{aligned}
\mathbb{E}[U] &= \frac{48}{50}\left(1 \cdot \frac{42}{44} + 20 \cdot \frac{1}{44} - 100 \cdot \frac{1}{44}\right) \\
&+ \frac{1}{50}\left(1 \cdot \frac{43}{44} - 100 \cdot \frac{1}{44}\right) \\
&+ \frac{1}{50}\left(1 \cdot \frac{43}{44} + 20 \cdot \frac{1}{44}\right) \\
&= -\frac{19}{22} \cdot \frac{48}{50} - \frac{57}{44} \cdot \frac{1}{50} + \frac{63}{44} \cdot \frac{1}{50} \\
&= -\frac{909}{1100} = -0.826
\end{aligned}$$

(c) The difference between the expected utilities, $-0.826 - (-0.64) = -\frac{41}{220} = -0.186$, is negative. Thus it is not better to switch the box and the value of perfect information is 0.

(d) The ability to chose between the revealed prize and opening another box means we gain a superpower: we can chose the option with the greatest expected utility. Mathematically, we express this by taking the maximum:

$$\max\left\{\mathbb{E}[U_r], \mathbb{E}[U_a]\right\}$$

where $\mathbb{E}[U_r]$ is the expected utility of keeping the revealed prize and $\mathbb{E}[U_a]$ is the expected utility of choosing choosing another box.

If you initially chose a box with a single bonus point ($p = \frac{48}{50}$), then $\mathbb{E}[U_r] = 1 \cdot 1$ and $\mathbb{E}[U_a] = 1 \cdot \frac{47}{49} + 20 \cdot \frac{1}{49} - 100 \cdot \frac{1}{49} = -\frac{33}{49}$. Now $\max\{\mathbb{E}[U_r], \mathbb{E}[U_a]\} = 1$ which is the expected utility of keeping the revealed prize. Thus when we initially choose the bonus point box it is best to keep the revealed prize because that will yield a higher expected utility.

If you initially chose a box with the free A on the final ($p = \frac{1}{50}$), then $\mathbb{E}[U_r] = 20 \cdot 1$ and $\mathbb{E}[U_a] = 1 \cdot \frac{48}{49} - 100 \cdot \frac{1}{49} = -\frac{52}{49}$. Now $\max\{\mathbb{E}[U_r], \mathbb{E}[U_a]\} = 20$ which is the expected utility of keeping the revealed prize. Thus when we initially choose the free A box it is best to keep the revealed prize because that will yield a higher expected utility.

If you initially chose a box that forces you to retake the course ($p = \frac{1}{50}$), then $\mathbb{E}[U_r] = -100 \cdot 1$ and $\mathbb{E}[U_a] = 1 \cdot \frac{48}{49} + 20 \cdot \frac{1}{49} = \frac{68}{49}$. Now $\max\{\mathbb{E}[U_r], \mathbb{E}[U_a]\} = \frac{68}{49}$ which is the expected utility of choosing another box. Thus when we initially choose the course retake box it is best to chose another box because that will yield a higher expected utility.

Putting everything together, we have

$$\begin{aligned}
\mathbb{E}[u] &= \mathbb{P}[U = 1] \cdot (\text{max utility btwn keeping bonus point or choosing again}) \\
&+ \mathbb{P}[U = 20] \cdot (\text{max utility btwn keeping A on final or choosing again}) \\
&+ \mathbb{P}[U = -100] \cdot (\text{max utility btwn keeping retake course or choosing again}) \\
&= \frac{48}{50} \cdot 1 + \frac{1}{50} \cdot 20 + \frac{1}{50} \cdot \frac{68}{49} \\
&= \frac{68}{49} = 1.388
\end{aligned}$$

Problem 3: Search Practice (20 points)

(a) DFS expands $\{S, A, C\}$ and returns $\{S, A, C, G\}$. BFS expands $\{S, A, B, D\}$ and returns $\{S, D, G\}$.

(b) DFS expands $\{S, A, C\}$ and returns $\{S, A, C, G\}$. BFS expands $\{S, A, B, D, C\}$ and returns $\{S, D, G\}$.

(c) UCS expands $\{S, A, B, D, C\}$ and returns $\{S, A, C, G\}$. A* expands $\{S, A, D, B, C\}$ and returns $\{S, A, C, G\}$.

(d) A suboptimal solution is $\{S, D, G\}$, which may be returned if C is never expanded. This can occur if the heuristic for C is larger than 4, which is inadmissible since the true cost from C to goal is 2. Given this, A* expands $\{S, A, D, B\}$ and returns $\{S, D, G\}$.

(e) To be admissible, $h(S) \leq 8$. To be consistent, $h(S) \leq 4$. The heuristic for S does not affect A* since it is always the first node to be inserted and then popped; there are no other nodes to compare with.

Problem 4: Word Ladders (50 points)

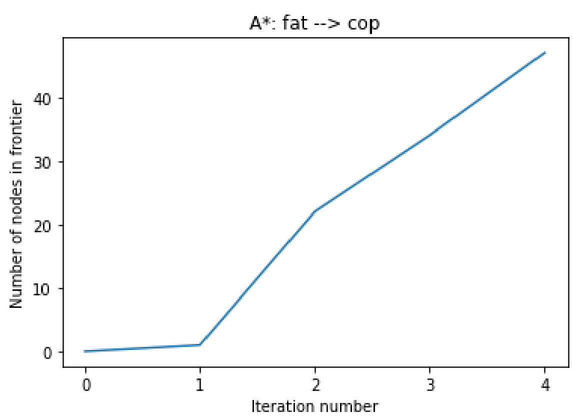
4.4 Analysis (20 points)

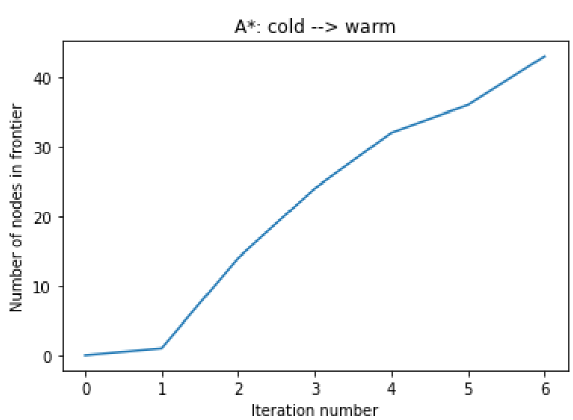
(a) (i) "fat" → "cop":
- Path: `['fat', 'cat', 'cap', 'cop']`
- Solution length: 4
- Solution path cost: 3
- Number of nodes expanded: 3

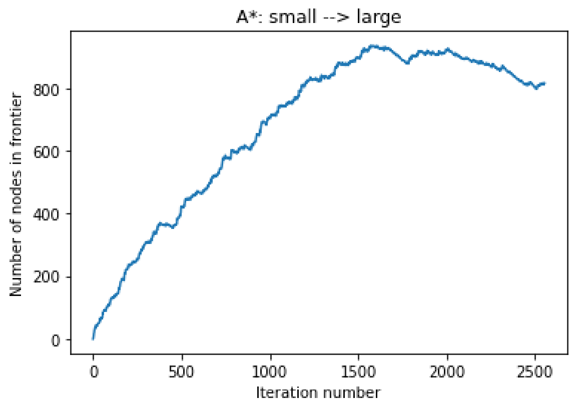
(ii) "cold" → "warm":
- Path: `['cold', 'cord', 'card', 'ward', 'warm']`
- Solution length: 5
- Solution path cost: 4
- Number of nodes expanded: 5

(iii) "small" → "large":
- Path: `['small', 'shall', 'shale', 'shade', 'shads', 'sheds', 'seeds', 'sends', 'sands', 'sandy', 'randy', 'rangy', 'range', 'mange', 'marge', 'large']`
- Solution length: 16
- Solution path cost: 15
- Number of nodes expanded: 2552

Plots:







(b) Any maximum depths from 4 to 14 (inclusive) would yield solutions for the first two puzzles but not the third. These solutions may not be identical to those of A*.

(c) Note that, below, "solution length" refers to the length of the list returned by `sequence()`, which includes both the start and goal nodes.

(i) `max_depth=4`

- "fat" → "cop": Solution length = 4, number of nodes expanded = 217.
  Same solution length as A*, but much more nodes expanded, as expected for the more inefficient DFS backbone of iterative deepening search.
- "cold" → "warm": Solution length = 5, number of nodes expanded = 402.
  Same solution length as A*, but much more nodes expanded.
- "small" → "large": No solution found, number of nodes expanded = 64.
  A* informed us that the solution is at depth 15, so a max depth of 4 results in no solution.

(ii) `max_depth=8`

- "fat" → "cop": Solution length = 4, number of nodes expanded = 217.
  Same solution length as A*, but much more nodes expanded, as expected for the more inefficient DFS backbone of iterative deepening search.
- "cold" → "warm": Solution length = 5, number of nodes expanded = 402.
  Same solution length as A*, but much more nodes expanded.
- "small" → "large": No solution found, number of nodes expanded = 2280.
  A* informed us that the solution is at depth 15, so a max depth of 8 results in no solution.

(d) (i) `max_depth=4`

- "fat" → "cop": Solution length = 5, number of nodes expanded = 124.
  Solution path is more costly than that found by IDS, because IDS first attempts smaller max depths. Number of nodes expanded is less than in IDS due to lack of repeating expansions at lower depths and directly finding a solution at depth 4.
- "cold" → "warm": Solution length = 5, number of nodes expanded = 286.
  Same solution length as IDS, fewer nodes expanded. The results from IDS and DFS are similar here, because the best (IDS) solution is at depth 4, so DFS searching directly with max depth of 4 isn't suboptimal.
- "small" → "large": No solution found, number of nodes expanded = 44.
  A* informed us that the solution is at depth 15, so a max depth of 4 results in no solution. Number of nodes expanded is fewer than in IDS because the iterations with smaller max depths do not occur in DFS.

(ii) `max_depth=8`

- "fat" → "cop": Solution length = 9, number of nodes expanded = 78.
  Solution path is more costly than that found by IDS, because IDS first attempts smaller max depths. Number of nodes expanded is less than in IDS due to lack of repeating expansions at lower depths and directly finding a solution at depth 8.

- “cold” → “warm”: Solution length = 9, number of nodes expanded = 52.
  Same explanation as “fat” → “cop”.
- “small” → “large”: No solution found, number of nodes expanded = 1199.
  A* informed us that the solution is at depth 15, so a max depth of 8 results in no solution. Number of nodes expanded is less than in IDS because the iterations with smaller max depths do not occur in DFS.