Sneh Shah (sjs413) - Section 2

Rishi Shah (rrs141) - Section 5

Intro to AI Project 3

()

April 10, 2021

Certification of Academic Integrity

I certify that the work done on this project is my own and is not copied or taken from online or any other student's work. - Rishi Shah
I certify that the work done on this project is my own and is not copied or taken from online or any other student's work. -Sneh Shah

Breakdown of Work

Rishi coded the basic agent functions. He worked on questions 1 and 3 of the write-up.

Sneh coded the advanced agent functions. She worked on questions 2 and 4 of the write up.

We worked together to come up with the pseudocode for all functions together and split up the coding as mentioned above.

Question 1.

P(Target Is Found In Cell i| Observations t and Failure in Cell j) =

$$\frac{\text{Belief State[i][j]} * \text{(False Negative Rate)}}{\text{P(Observations t and Failure in Cell j)}}$$

Bayes Theorem can be used to efficiently update the belief state with the equation above. When a cell results in a failed search, its current probability, Belief State[i][j], is multiplied by the probability of failure/false negative rate (which is dependent on terrain type). After this, the total probability on the board is retrieved by summing up all the observations up to time t. Then, all the cells on the board are divided by this value to normalize the board and make sure all probabilities add up to 1.

Question 2.

The probability that the target is found in Cell i is equivalent to:

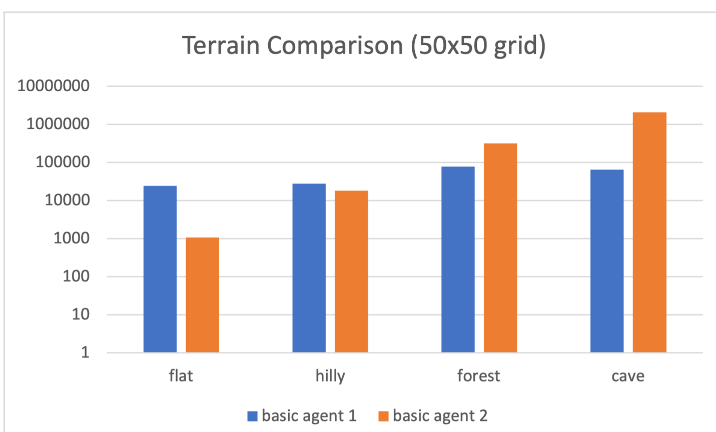
P(Target Is Found In Cell i| Observations t) =

$$\frac{\text{Belief State[i][j]} * \text{(1-False Negative Rate)}}{\text{P(Observations t)}}.$$

Belief State[i][j] is the current probability of cell (i,j) and (1-False Negative Rate) is the probability of successfully finding the target in a given terrain. We need to multiply these two probabilities because in order to find a target in a cell, we need the probability that the target is in the cell and the probability that the target is found in the cell. Then we divide by P(Observations t) which is the sum of Belief State for each cell.

Question 3.

The following graph depicts the difference between basic agent 1 and basic agent 2. The graph is on a logarithmic scale due to the large differences in scores for each agent.

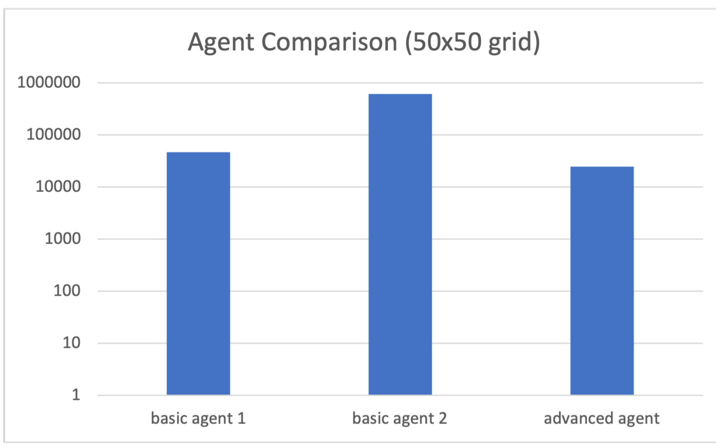


	flat	hilly	forest	cave
basic agent 1	24165.53	27789.39	77650.36	65858.18
basic agent 2	1085.63	18134.2	318225.7	2114627

On average, we can see that basic agent 1 is better than basic agent 2. However, the performance of the agent can depend on terrain. Taking a look at our terrain analysis results above, we can see that agent 2 is better for the terrains with a lower false negative rate like flat and hilly. On the other hand, basic agent 1 is significantly better than basic agent 2 on the terrains with higher false negatives rates like forest and caves. Basic agent 2 does better in situations where the target is in a flat or hilly cell because basic agent 2 prioritizes cells that are easier to search and more likely to be successful.

Question 4.

The following graph depicts the difference between basic agent 1, basic agent 2, and advanced agent. The graph is on a logarithmic scale due to the large differences in scores for each agent. The graph shows us that advanced agent performs better than basic agent 1 and basic agent 2 on average.



basic agent 1	46407.05
basic agent 2	608816.14
advanced agent	24600.3

For our advanced agent, we consider how far away the cell with the highest probability is. For basic agents 1 and 2, we travel to the cell with the highest probability, (belief state), and add the Manhattan distance to our total score. However in advanced agent, we check if there is any cell worth searching on the way to the cell with the highest probability. For example, if we are currently at the cell (0,0) and the cell with the next highest probability is (0,5), basic agents 1 and 2 would essentially teleport to (0,5). Our advanced agent would check if there are any cells in the path from (0,0) to (0,5) worth searching.

We determine the utility of directly travelling to the cell with the highest probability and the utility of searching one of the current cell's neighbors. The utility of travelling directly to the cell with the highest probability is:

baseValue = baseDiscount * belief[(newX, newY)]. The baseDiscount = 1/(distX+distY+1),

where distX and distY are the distances from the current (x,y) to the (x,y) with the highest probability and we add 1 to represent searching that cell. This basically represents the inverse of the Manhattan distance, causing the closer square to be weighed more. belief[(newX, newY)] is the probability of the cell with the highest probability.

The utility of searching a neighbor is:

value = discount * belief[(i, j)]. The discount = 1/(distNewX+distNewY+1),

where distNewX and distNewY are the distances from the current cell's neighbors distance to the cell with the highest probability and again, add 1 to represent searching the cell. belief[(i,j)] represents the probability of the neighbor. We determine the utilities of all neighbors and if the max utility is greater than baseValue, then we move to that cell and search. If there are no values that are greater than base value, that means that there is no current neighbor worth searching. Instead, we pick a neighbor that brings us closer to the cell with the highest probability, move to that cell, and don't search it, and repeat the process of calculating utilities.

For example, if we are at (0,0) and the cell with the highest probability is (0,5) and the probability is 0.5, advanced agent would check (0,0)'s neighbors and calculate the utilities. The utility of directly travelling to (0,5) would be 1/(0+5+1) * 0.5 = 0.08. Let's say (0,1) has 0.45 as it's probability and (1,0) has 0.05 as it's probability. The utility for (0,1) would be 1/(0+4+1) * 0.45 = 0.09 and the utility of (1,0) would be 1/(1+5+1) * 0.05 = 0.007. The max utility for the neighbors would be (0,1)'s utility of 0.09 and since it is greater than 0.08, we would move to this cell and search it.

The advanced agent performs better than both basic agent 1 and basic agent 2 because it limits the distance traveled. In the example mentioned above, basic agents 1 and 2 would go to (0,5) and if (0,5) returned unsuccessful, it might go all the way back to (0,1) since it has a high probability. This adds to the total distance traveled and thus add to the total score. However, advanced agent notices that 0.45 is a high probability and that it might be worth searching this cell on it's way to (0,5).

Given enough time and resources, the advanced agent could find the best possible path from its current location to the location of the highest probability. This "best path" would represent the path that has the highest chance of finding the target and can be determined by using BFS to look through all possible paths and seeing the average probability on each path. The path with the highest average probability is the one that the agent would travel on and any cell on that path that had a probability higher than that average probability can be searched.

5. Clever Acronym

basic agent 1: CTS (contains target search)

basic agent 2: FTS (finds target search)

advanced agent: WITS (worth it target search)