Assignment 3: Probabilistic Search (and Destroy)

Karun Kanda (Section 7), Agam Modasiya (Section 3), and Varun Gaddam (Section 3)

April 10th, 2021

1 Usage

To run Basic Agent 1:

```
python Run_Agent.py ba1
```

To run Basic Agent 2:

```
python Run_Agent.py ba2
```

To run Advanced Agent:

```
python Run_Agent.py adv
```

2 Language/Libraries Used

For this project we used Python and the Python libraries we used were:

1. numpy - To create nice nested arrays to represent the map with the respective terrains and the agent's belief state and confidence state.
2. random - To obtain the random number when picking between distances, when deciding if the agent found the target, etc.
3. sys - To get command line arguments to dynamically create the map, belief state, and confidence state.

3 Questions about Project 3: (Analysis)

3.1 Given observations up to time t (Observations$_t$), and a failure searching Cell$_j$ (Observations$_{t+1}$ = $Observations_t \wedge$ Failure in $Cell_j$), how can Baye's theorem be used to efficiently update the belief state? i.e., compute:

$$\mathbf{P(\text{Target in Cell}_i|\ \text{Observations}_t \wedge \text{Failure in Cell}_j).}$$

Given Baye's theorem that $P(A|B)P(A) = P(B|A)P(A)$ we can use that to break down P(Target in Cell$_i$| Observations$_t$ $\wedge$ Failure in Cell$_j$) to:

$$\frac{P(\text{Failure in Cell}_j|\text{Target in Cell}_i)P(\text{Target in Cell}_i|\text{Observations}_t)}{P(\text{Failure in Cell}_j)}$$

Then once we get this from Baye's theorem the numerator can be broken down to probabilities that we already know. First lets take P(Failure in Cell$_j$| Target in Cell$_i$). For this probability there are two cases we should handle for this when $i = j$ and when $i = / = j$. Before we get to that, lets consider the i^{th} cell to be any other cell in the belief state and the j^{th} cell to be the current cell that we just looked at. Now taking that into consideration, lets consider the case when $i = j$ or when the cell we are updating is equal to the cell we just looked at. In that case, we have a failure in the current cell but we are given the information that the target which in fact is equivalent to the probability of the target not being found in Cell i given the target is in cell i. Which we already know because it is given to us. So P(Failure in Cell$_j$|Target in Cell$_i$) is just the false negative rate of the current cell (Cell$_i$). Now lets consider the second factor in the numerator P(Target in Cell$_i$|Observations$_t$). This is simply given to us to because if we recall when we calculate for the belief[Cell$_i$] that is equal to P(Target in Cell$_i$|Observations$_t$) so we can substitute P(Target in Cell$_i$|Observations$_t$) for belief[Cell$_i$]. Which makes the numerator for $i = j$ False Negative Rate of Cell$_i$ * belief[Cell$_i$]. Now for the denominator we decided to break down using marginalization to break down how many ways the current cell j can fail. So to break it down, Cell j can fail when we reach the target meaning if we reached the target but we hit a false negative rate or if we straight up are not at the target. Which gets us to the denominator being the false negative rate at cell$_j$ * belief[Cell$_j$] + (1− belief[Cell$_j$]). Now that we have the numerator and denominator the full equation in terms of knowledge we have ends up being:

$$\frac{\text{False Negative Rate of Cell}_i * \text{belief[Cell}_i]}{\text{false negative rate at cell}_j * \text{belief[Cell}_j] + (1 - \text{belief[Cell}_j])}$$

Now that we found the equation for when $i = j$ we can apply the same logic for when $i = / = j$. Like for instance in the numerator P(Target in Cell$_i$|Observations$_t$) is still going to be belief[Cell$_i$] and in the denominator P(Failure in Cell$_j$) is still false negative rate at cell$_j$ * belief[Cell$_j$] + (1− belief[Cell$_j$]). However in the numerator P(Failure in Cell$_j$|Target in cell$_i$) turns out to be different because we are working with failure happening in the current cell but given the target can be in any of the other cells. Which is the case that it is guaranteed to happen because if the target is in the other cells than the current cell is bound to fail due to $i = / = j$. So we get P(Failure in Cell$_j$|Target in cell$_i$) to be just 1. Now with this information we can formulate teh equation to be:

$$\frac{\text{belief[Cell}_i]}{\text{false negative rate at cell}_j * \text{belief[Cell}_j] + (1 - \text{belief[Cell}_j])}$$

Therefore using Baye's theorem we utilize it to put the equation in terms of knowledge we are aware of to efficiently update the belief state in terms of when $i == j$ and $i = / = j$:
When $i == j$,

$$\frac{\text{False Negative Rate of Cell}_i * \text{belief[Cell}_i]}{\text{false negative rate at cell}_j * \text{belief[Cell}_j] + (1 - \text{belief[Cell}_j])}$$

and when $i =/= j$,

$$\frac{\text{belief[Cell}_i]}{\text{false negative rate at cell}_j * \text{belief[Cell}_j] + (1 - \text{belief[Cell}_j])}$$

3.2 Given the observations up to time t, the belief state captures the current probability the target is in a given cell. What is the probability that the target will be found in Cell i if it is searched:

$$\textbf{P(Target found in Cell}_i| \textbf{ Observations}_t\textbf{)?}$$

To find P(Target found in Cell_i| Observations_t) we can first approach to finding this probability by breaking down target found in Cell_i to two pieces. For this event, we know that we can break it up into two seperate events: Target being in Cell_i and Success in Cell_i. Knowing this information we can rewrite the probability into P(Target in $\text{Cell}_i \wedge$ Success in $\text{Cell}_i|\text{Observations}_t$). Now with this break down, we know Success in Cell_i is independent of Observations_t because we found the cell with the target so the observations at the point are meaningless. Also because the conjunction we can break the problem further into P(Target in $\text{Cell}_i \wedge$ Success in $\text{Cell}_i|\text{Observations}_t$) to P(Target in $\text{Cell}_i|\text{Observations}_t$) * P(Success in $\text{Cell}_i|\text{Observation}_t$). Now with the knowledge that success in the current cell is independent of observations_t, we rewrite the current formula to P(Target in $\text{Cell}_i|\text{Observations}_t$) * P(Success in Cell_i). Now with the new equation, we can utilize knowledge that we know already and incorporate that to solve it. P(Target in $\text{Cell}_i|\text{Observations}_t$) = belief[$\text{Cell}_i$]. Then P(Success in Cell_i) = 1 - False Negative Rate of the current cell because success implies you find the target. However, when you reach the target there is always the chance you hit a false negative and you don't find the target which would result to a failure. Knowing this you find success by applying the complement rule where P(A)+P(A')=1 (where even A represents failure and A' represents success) by setting P(Success in Cell_i) = 1 - False Negative Rate of Cell_i. Which we get P(Target found in $\text{Cell}_i|\text{Observations}_t$) = belief[$\text{Cell}_i$] * (1 - False Negative Rate of Cell_i). Therefore the probability that the target will be found in Cell i if its searched is belief[Cell_i] * (1 - False Negative Rate of Cell_i).

3.3 Generate 10 maps, and play through each map (with random target location and initial agent location each time) 10 times with each agent. Which agent is better, on average?

For this test we generated 10 boards with the target terrain in flat and located in a random location in the map. Then made the basic agents play on that map 20 times. Once that was done we repeated the process for the other 3 terrains. Once we conducted those tests we were left with 20 trials on each individual map which we took the average over 20 trials to find the average score for each terrain and then took the average of the 10 boards. Here are the results from the given tests:

	Flat	Hilly	Forested	Caves
Basic Agent 1	2807.5	2161.78	3233.69	17522.03
Basic Agent 2	1429.17	1176.345	3275.16	16903.74

Table 1: Average Score per Terrain

From the table we can see that basic agent 2 was able to out perform basic agent 1 in all terrain types except for forested making basic agent 2 better than basic agent 1 on average. Based on the way we designed basic agent 2 it picks what cells to go to based on a confidence level (based on problem 2) that provides the agent the probabilities the target will be in Cell_i and be found if it is searched which gets updated right after the belief state is updated. With this new state of confidence, the agent is able to make a better prediction for where the target is because we take an account for where there is more of a chance of the agent succeeding to find the agent rather than just containing the target. This is why basic agent 2 was able to out perform basic agent 1 for lower false negative rates. Basic agent 1 out performs basic agent 2 when the target's terrain is forested. The main reason is we get to a point where the false negative rate takes an effect on getting to the target. For the forested terrain the false negative rate is 70% and if we consider the case when the queried cell is the target, there is only a 30% chance that we actually find the target when we land on the target because the false negative rate is so high.

3.4 Design and implement an improved agent and show that it beats both basic agents. Describe your algorithm, and why it is more effective than the other two. Given world enough, and time, how would you make your agent even better?

As we learned from the basic agents, if the target is in a cell with lower false negative rate it easier to find if we use the confidence levels. But agent 1 out performs agent 2 as the false negative rates go up. We concluded that in the beginning it would be good to rely on the confidence levels and as we more searches we should rely more on the beliefs and expected future states. The advanced agent combines both basic agent 1 and 2. In the beginning when the agent is starting to do the searches, it behaves like agent 2, where the confidence level is given higher priority. But as the agent does more and more searches, there is a higher chance that the target is in a cell with lower confidence levels. To counteract that we decrease the importance of the confidence levels in relationship with the number of cells searched. Hence why the advanced agent out performs basic agent 1 and basic agent 2.

	Flat	Hilly	Forested	Caves
Basic Agent 1	2807.5	2161.78	3233.69	17522.03
Basic Agent 2	1429.17	1176.345	3275.16	16903.74
Advanced Agent	377.1	676.505	3134.35	16305.99

Table 2: Average Score per Terrain

We came up with a set of rules for the searches that is used in the advanced agent algorithm:

1. At every cell the agent starts by searching all of it's neighbors.
2. If the terrain of a cell is flat, the agent searches twice. The agent does this because there is only a 1% chance that the target won't be discovered, given it's in the cell.
3. To look for which cell to search next, the agent goes to each cell and calculates the belief if the search were to fail for that cell and all six of it's neighbors. It also re-calculates the confidence of finding the target, given it's in that cell. The beliefs and confidence levels are then summed together. The cell with highest sum is chosen to be searched next. If there are 2 cells that have equal sum, the cell with smaller distance is chosen.
4. When the agent is going from current cell to the next one, it also looks for cells to search on it's way. If there is a cell that has a higher belief that (0.85 x belief of the next cell), it is searched and the probabilities are updated accordingly.
5. Lastly, as more and more searches are done, the importance of confidence level decreases. To account for that, when the confidence levels are added each of them is multiplied by an alpha value. The alpha value is (numbers of cells / (numbers of cells searched + (number of cells x 2)). As the number of cells searched increases, the value of alpha value decreases.

If we had enough time, we would go more time steps ahead and calculate the beliefs of cells and their confidence. We can then use those values to search for cells that have the higher belief and confidence after the specified time steps. This would help us decrease the amount of steps we need to take and the cells we need to search.

4 Contributions and Acknowledgments about Rutgers Academic Policy

Group Member: Karun Kanda **Netid:** kk951 **RUID:** 184007619

For the final writeup I worked answering questions 1 and 2 making the solutions for both probabilities and question 3 analyzing/comparing basic agent 1 versus basic agent 2.

I, Karun Kanda, certify that the work submitted in this project is my own work and not copied or taken from online or any other student's work.

Group Member: Varun Gaddam **Netid:** vrg24 **RUID:** 187008136

I worked on question 3 related to the basic Agent 1. I assisted in testing the data related to comparing the three agents.

I, Varun Gaddam, certify that the work submitted in this project is my own work and not copied or taken from online or any other student's work.

Group Member: Agam Modasiya **Netid:** ajm432 **RUID:** 185009911

I worked on doing question 4 and helped with getting the data. I also helped with proofreading

I, Agam Modasiya, certify that the work submitted in this project is my own work and not copied or taken from online or any other student's work.