RUTGERS UNIVERSITY - DEPARTMENT OF COMPUTER SCIENCE

SEARCH AND DESTROY

Khanh Nguyen
April 9, 2021

1 ABSTRACT

In this report, we directly model our knowledge/belief about a system probabilistic, and use this belief to efficiently direct the future.

2 REPRESENTATION

2.1 CODE

In our code, the map is represented as a numpy matrix. The value assigned to the numpy matrix are randomly assigned based on the probability of the terrain type (flat with probability of not founding 0.1, hilly with probability of not founding 0.3, forested with probability of not founding 0.7, and lastly caves with probability of not founding 0.9). Our code has terminal based input variables through which we can change the map size (default 10 * 10), choose if we want to visualize the map, change the decision rule (Rule 1 or Rule 2) etc.

2.2 VISUALIZATION

Figure 1 displays the visualization of the map. Note that the map is represented as heat map. The four parts in the figure as explained below:

- The left top map shows the actual map of the terrains generated by the code. The red symbol, X shows the target location in the map. The color range of the actual map from the lightest to the darkest correspond to Flat, Hilly, Forested and Cave

- The right top map shows the belief matrix, which basically gives the probability of target being in a particular cell.
- The bottom left shows the confidence matrix(probability of target being found in a cell) as our agent opens up cells based on our observations made. The shade of belief and confidence matrix determines the probability in that cell. In both these cases, as the probability increases the shade of color increases in the heat map.
- The bottom right map shows the cells that the agent checks. The more the cell has been opened, darker the shade.

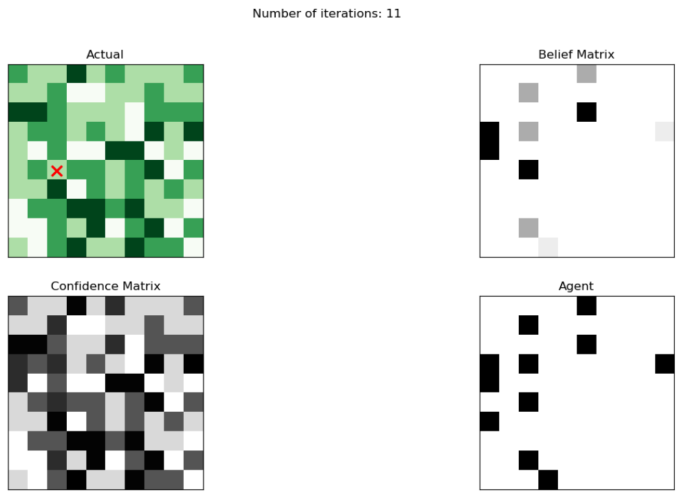

Figure 1: Probabilistic Hunting Visualisation

3 A STATIONARY TARGET

3.1 HOW TO EFFICIENTLY UPDATE BELIEF STATE?

The information at any given time t is the observations till time t and the failure of searching the current cell. Updating the belief state is the same as finding the probability of target in a new cell given that we know the observations till now and the failure in the previous cell. Using this information, the agent updates the belief state. The agent works in one of the three states - Initial, Update and Termination.

- **Initial State:** This state occurs only when t = 0. At this state the agent assigns equal probability to each of the cells. This is because at t = 0, there are no observations and the failure of searching the previous cell is not there. At this state, the agent opens the cell with maximum probability (which in this case is a random cell) and it moves to the update state.

- **Update state:**The agent stays in the update state until it is able to find the target.Note that this is the state where the agent spends the most time. In this state at any given time t two recurring steps occur.
  - **Step 1:** The agent checks if the cell opened at time t has the target. If the target is present, then the agent moves to the termination state. If the target is not present,then the agent moves to Step 2.
  - **Step 2:**The agent reduces the probability of the current cell having the target by some factor (which is the false negative rate of the current cell's terrain). Then, as the overall probability is not equal to one, the agent normalizes all the cells. For example,let us consider a map size of 3 3 and the current cell is terrain Flat (Note that the false negative rate of the terrain flat is 0.1.), then,
    - Inital probability of current cell = 1/9
    - On opening current probability of current cell = 0.1 * 1/9 = 1/90
    - The total changes after the agent opens the current cell
      - Before opening the cell, total probability: 1/9 x 9 = 1
      - After opening the cell, total probability: 1 * 1/90 + 8 * 1/9 = 81/90
    - The agent then normalizes the value
      - Probability of current cell: 1/90 / 81/90 = 1/81
      - Probability of remain 8 cells: 1/9 / 81/90 = 10/81
  - Now, the total probability adds up to 1. Then, the agent goes back to Step 1 and it recursively runs.
- **Termination stage:** The agent comes to this state from Step 1 of the update state. In this state, the agent generates a random number between 0 and 1. If the randomly generated number is greater than the false negative rate of the current cell (target), then the agent is sure that the target exists in the current cell and target is found. If not, then the agent goes back to Step 1 of the update state and continues.

3.2 How to efficiently update the confidence state?

The information at given time t is the observations till time t. The confidence state is updated only after the belief state is updated. A confidence state is a cell on the map. Each cell is of a certain terrain (flat, hilly, forest or cave). A confidence matrix comprises of the cells on the map. A cell in the confidence matrix is updated using the following formulae. where, Confidence State[i, j] is the confidence state of cell of i^{th} row and j^{th} column, FNR(Terrain) is the false negative rate of the terrain present in the cell and Belief State[i, j] is the belief state of cell of i^{th} row and j^{th} column after the update.

3.3 Comparing the two decision rules

The two rules that are compared are :

- **Rule 1:**At any time, search the cell with the highest probability of containing the target
- **Rule 2:** At any time, search the cell with the highest probability of finding the target

In case of Rule 1, the decision to search the cell highest probability of containing the target is made based on the belief state after every iteration. Whereas, in case of rule 2, the decision to search the cell with the highest probability of finding the target is made based on the confidence state after every iteration. Note that the confidence state is updated only after the belief state is updated every iteration.

On average, Rule 2 requires lesser searches when the target is present in terrains other than cave and, hence performs better. But, when the target is present in the cave terrain, rule 1 on average performs better. This behavior is seen in Figure 2 when the map size is fixed at 10.

This behavior is seen because rule 2 in general, has more information than rule 1 at any given time. Rule 2 uses the information present in the belief state (after its update) along with the information of the terrain (False Negative Rate) before making a decision. Therefore, when the target is present in terrains beside a cave, the rule 2 is able to find the target sooner compared to rule 1. When the target is present in the cave terrain, rule 2 does not easily find the target because its False Negative Rate (FNR) is quite high (0.9). This high FNR causes only a small increase in confidence over time and therefore, rule 2 takes more time when compared to rule 1.

Yes, the behavior of rules 1 and 2 hold across multiple maps as seen in Figure 3. You can observe that the trend of iterations for each terrain follows the same pattern.

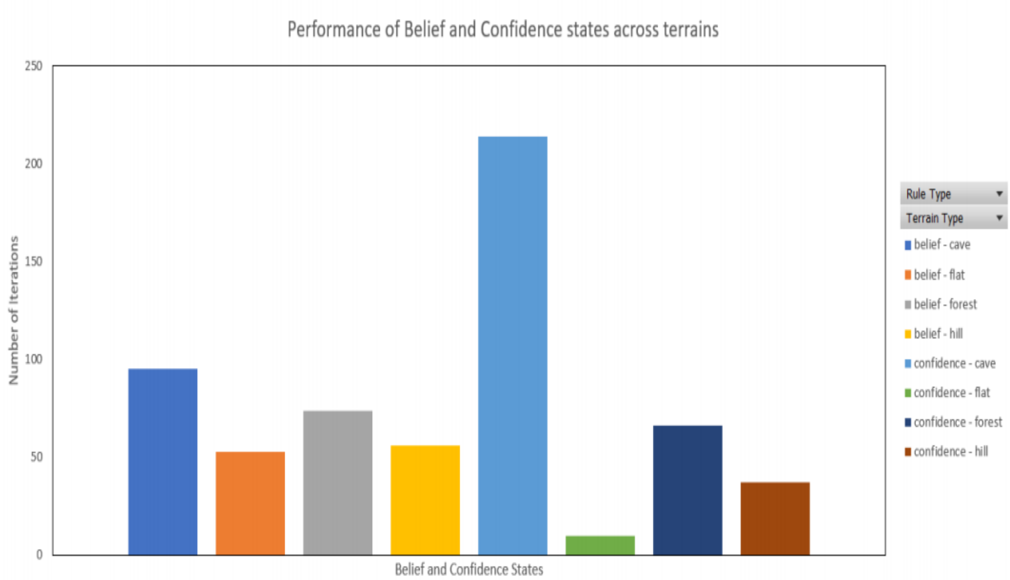

Figure 2: Comparison of the belief state and confidence state across terrains when map size = 10

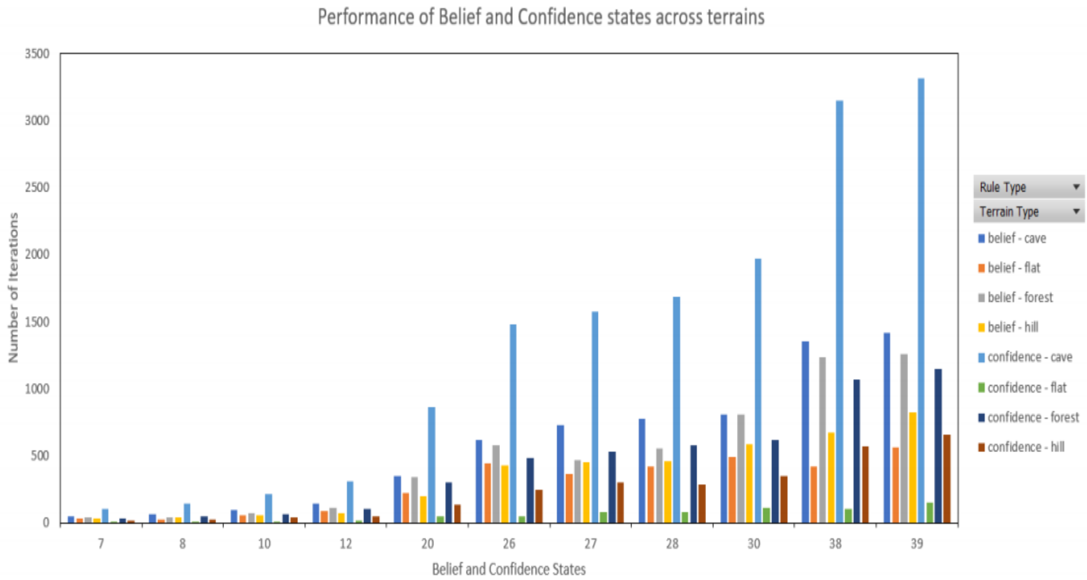

Figure 3: Comparison of the belief state and confidence state across terrains across different map sizes

3.4 What happens when distance is included?

3.4.1 How can you use the belief state and your current location to determine whether to search or move (and where to move), and minimize the total number of actions required?

The agent makes a decision based on two factors - current belief state and current location(where the agent is currently searching). Using this information, the agent can do one of two things. It can either search the current cell or move to another cell. Note that either task performed is counted as an action. The agent makes a decision in the following manner :

- **Initial state:** This state occurs only when t = 0. At this state the agent assigns equal probability to each of the cells. The agent opens a random cell and it moves to the update state.

- **Update state:** The agent stays in the update state until it is able to find the target. Note that this is the state where the agent spends the most time. In this state at any given time t three recurring steps occur.
  - **Step 1:** The agent checks if the cell opened at time t has the target. If the target is present, then the agent moves to the termination state. If the target is not present, then the agent moves to Step 2.
  - **Step 2:** The agent reduces the probability of the current cell having the target by some factor (which is the false negative rate of the current cell's terrain). Then,

as the overall probability is not equal to one, the agent normalizes all the cells. In addition to this, the agent also computes the Manhattan distance from the current cell to all other cells. The probability computed in each cell is divided by its Manhattan distance from the current cell. Then the agent moves to Step 3.

  - **Step 3**: The agent moves to the cell with the highest value. Note that this cell is the closest cell that is likely to have the target from the current cell. The agent moves to the chosen cell (the one with the highest value). The agent moves to Step 1.

- **Termination State:** The agent comes to this state from Step 1 of the update state. In this state, the agent generates a random number between 0 and 1. If the randomly generated number is greater than the false negative rate of the current cell (target), then the agent is sure that the target exists in the current cell and target is found. If not, then the agent goes back to Step 1 of the update state and continues.

3.4.2 Compare performance of Rule 1 and Rule 2 based on new decision rules

The two rules that are compared are :

- **Rule 1:** At any time, search the cell with the highest probability of containing the target
- **Rule 2:** At any time, search the cell with the highest probability of finding the target.

In case of Rule 1, the decision to search the cell highest probability of containing the target is made based on the belief state after incorporating the distance factor (after every iteration). Whereas, in case of rule 2, the decision to search the cell with the highest probability of finding the target is made based on the confidence state after every iteration. Note that the confidence state is updated only after the belief state is updated every iteration.

Figure 4 shows the comparison of performance of rule 1 and 2 with/without distance. On average, it is seen that rule 1 does better with the additional information of current location and distance. But, in rule 1, if the target is present in the terrain - cave, the agent takes longer to find (due to the FNR). On the other hand, on average, the additional distance factor does not have a big impact on rule 2. Similar to rule 1, if the target is present in the cave terrain, the agent takes longer to find with the additional information.

This behavior is seen because the additional information of distance incorporated (with a low FNR) helps to narrow down on the target sooner. But, it is also seen that as the FNR increases the additional information slows down in finding the role. In rule 2, especially, there is no major difference because the majority role is played by FNR.

Note that the behavior of rules 1 and 2 hold across multiple maps as seen in Figure 5.

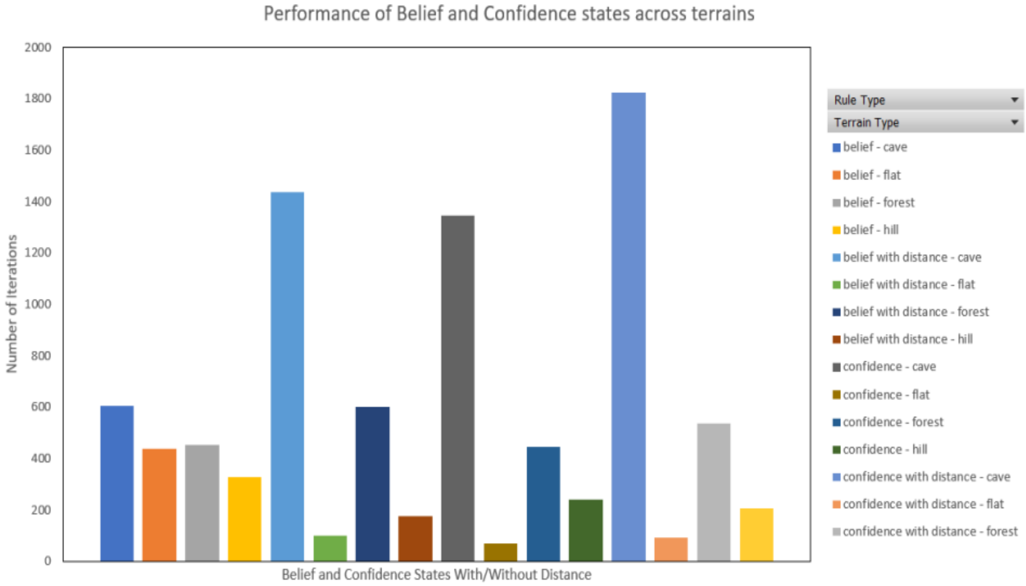

Figure 4: Comparison of the belief state and confidence state across terrains when map size = 25 with/without distance

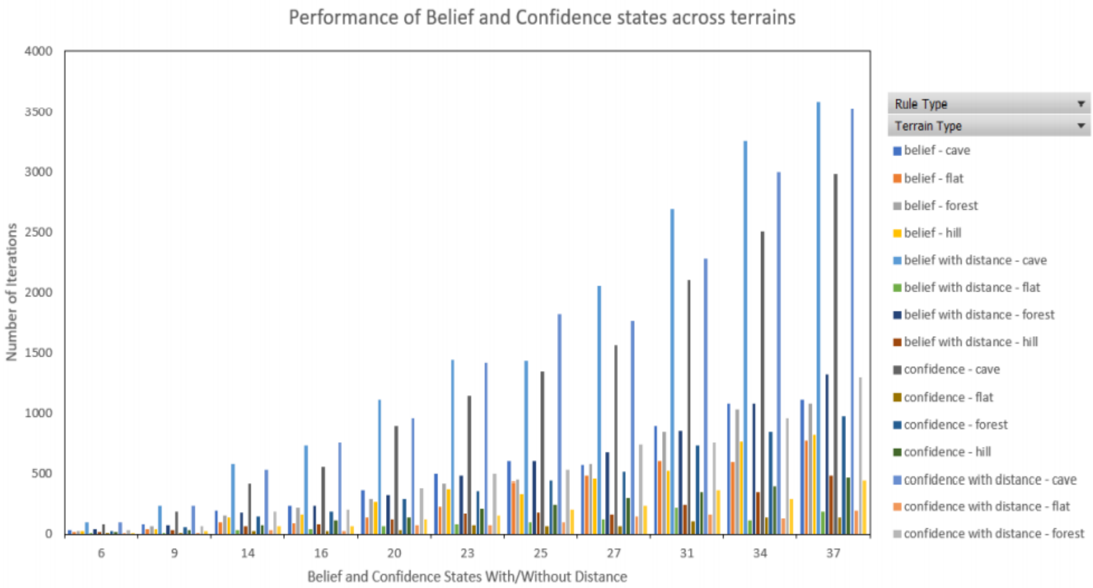

Figure 5: Comparison of the belief state and confidence state with/without distance accross different map sizes