Probabilistic Hunting Report

April 10, 2021

Group Information

Reagan McFarland (rpm141)

Alay Shah (acs286)

Toshanraju Vysyaraju (tv135)

Split

- Representation: Reagan, Alay, and Toshanraju
- Basic Agents: Reagan, Alay, and Toshanraju
- Improved Agents: Reagan, Alay, and Toshanraju
- Questions and Write Up: Reagan, Alay, and Toshanraju

Statements

- Reagan McFarland: On my honor, I have neither received nor given any unauthorized assistance on this assignment.
- Alay Shah: On my honor, I have neither received nor given any unauthorized assistance on this assignment.
- Toshanraju Vysyaraju: On my honor, I have neither received nor given any unauthorized assistance on this assignment.

Calculating Belief Cell i, j

Belief$[i, j] = P(\text{Target in } i, j|\text{Observations}_t)$

We can compute this recursively so essentially we just need to figure out the prob for the observation at time t , since we already know Belief$[i, j] = P(T_{i,j})_t$ at time t.

At time $t = 0$, Belief for all cells will be $1/|\text{cells}|$ i.e. $1/dim^2$

Since the observation will necessarily be a fail in cell i, we will only deal with this case. (if we find the target, we have no reason to continue)

Let $T_{i,j}$ be the event that the target is in cell i, j

Let $F_{i,j}$ be the event that the target is found in cell i, j

Known values:

$P(T_{i,j})$ at time $t - 1 \rightarrow B_{t-1}[i, j]$

$P(T_{i,j}|F_{i,j}) \rightarrow$ will always be 1, our false positive rate is 0

$P(\neg F_{i,j}|T_{i,j}) \rightarrow$ will be equal to the value given (.1 for flat, .3 for hill, .7 for forest, .9 for caves)

$P(F_{i,j}|T_{i,j}) = 1 - P(\neg F_{i,j}|T_{i,j})$

Now to calculate $P(T_{i,j}|\neg F_{i,j})$:

$P(T_{i,j}|\neg F_{i,j}) = \frac{P(T_{i,j} \wedge \neg F_{i,j})}{P(\neg F_{i,j})}$ using the Conditional Probability Formula

Numerator:

$P(T_{i,j} \wedge \neg F_{i,j}) \rightarrow P(T_{i,j})P(\neg F_{i,j}|T_{i,j})$ (conditional factoring)

We already know both these values from the given

Denominator:

$P(\neg F_{i,j}) = P(T_{i,j} \wedge \neg F_{i,j}) + P(\neg T_{i,j} \wedge \neg F_{i,j})$ (marginalization)

$P(\neg F_{i,j}) = P(T_{i,j})P(\neg F_{i,j}|T_{i,j}) + P(\neg T_{i,j})P(\neg F_{i,j}|\neg T_{i,j})$ (conditional factoring)

$P(T_{i,j})P(\neg F_{i,j}|T_{i,j}) \rightarrow$ All values are known

$P(\neg T_{i,j})P(\neg F_{i,j}|\neg T_{i,j}) \rightarrow [1 - P(T_{i,j})]P(\neg F_{i,j}|\neg T_{i,j}) \rightarrow [1 - P(T_{i,j})] * 1 \rightarrow [1 - P(T_{i,j})]$ (false pos 0)

$P(\neg F_{i,j}) = P(T_{i,j})P(\neg F_{i,j}|T_{i,j}) + [1 - P(T_{i,j})]$

All together:

$P(T_{i,j})_{t+1} = \frac{P(T_{i,j})P(\neg F_{i,j}|T_{i,j})}{P(T_{i,j})P(\neg F_{i,j}|T_{i,j})+[1-P(T_{i,j})]}$

This can be easily computed using the known values from above.

Problem 1: Given observations up to time t ($Observations_t$), and a failure searching Cell j ($Observations_{t+1} = Observations_t \wedge FailureinCell_j$), how can Bayes' theorem be used to efficiently update the belief state?

Again, since we know the beliefs are computed recursively, we just need to account for the current observation at time t

Updated belief given current observations up to time $t-1$ and fail at time t

For a given cell at k, l what is the belief

Known Values:

$P(T_{k,l})$ at time $t-1 \rightarrow$ belief of target being in k, l at time $t-1$

We have already calculated $P(\neg F_{i,j}) \rightarrow$ prob of observation (not finding target in i, j)

$P(\neg F_{i,j}|T_{k,l}) \rightarrow$ will always be 1, our false positive rate is 0 (prob of not finding target in i, j given the target is in k, l)

Now to calculate $P(T_{k,l})_{t+1}$:

$P(T_{k,l})_{t+1} = P(T_{k,l}|\neg F_{i,j}) = \frac{P(T_{k,l} \wedge \neg F_{i,j})}{P(\neg F_{i,j})}$ (Conditional Probability Formula)

Numerator:

$P(T_{k,l} \wedge \neg F_{i,j}) = P(T_{k,l})P(\neg Fi, j|T_{k,l}) \rightarrow P(T_{k,l}) * 1$ (conditional factoring and false positive rate = 0)

Denominator:

$P(\neg F_{i,j}) \rightarrow$ known from before

Putting this all together, we get $P(T_{k,l})_{t+1} = \frac{P(T_{k,l})}{P(T_{i,j})P(\neg F_{i,j}|T_{i,j})+[1-P(T_{i,j})]}$

This can be easily computed using the known values from above.

Problem 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?

Computing new probability for searched Cell i, j:

$P(T_{i,j} \wedge F_{i,j} | \neg F_{i,j}) = \frac{P(T_{i,j} \wedge F_{i,j}) * P(\neg F_{i,j} | T_{i,j} \wedge F_{i,j})}{P(\neg F_{i,j})}$ (Conditional Probability Equation)

Numerator:

$P(T_{i,j} \wedge F_{i,j}) * P(\neg F_{i,j} | T_{i,j} \wedge F_{i,j})$

When conditioning on where the target is, none of the other observations matter, since they are already accounted in the value of $T_{i,j}$. Thus, this allows us to simplify $P(\neg F_{i,j} | T_{i,j} \wedge F_{i,j})$ to $P(\neg F_{i,j} | T_{i,j})$.

Using conditional factoring, $P(T_{i,j} \wedge F_{i,j}) = P(T_{i,j}) P(F_{i,j} | T_{i,j})$

$= P(T_{i,j}) P(F_{i,j} | T_{i,j}) P(\neg F_{i,j} | T_{i,j})$

Denominator:

$P(\neg F_{i,j}) \rightarrow$ shown easy to compute

$= \frac{P(T_{i,j}) P(F_{i,j} | T_{i,j}) P(\neg F_{i,j} | T_{i,j})}{P(\neg F_{i,j})}$

$= P(F_{i,j} | T_{i,j}) * \frac{P(T_{i,j}) P(\neg F_{i,j} | T_{i,j})}{P(\neg F_{i,j})}$

$= P(F_{i,j} | T_{i,j}) * P(T_{i,j})_{t+1}$

We already showed how to compute $P(T_{i,j})_{t+1}$ and $P(F_{i,j} | T_{i,j})$ is a given probability, thus $P(T_{i,j} \wedge F_{i,j} | \neg F_{i,j})$ is easy to compute using the given values.

For $k, l \neq i, j$, we want to find $P(T_{k,l} \wedge F_{k,l} | \neg F_{i,j})$

We can break this down:

$P(T_{k,l} \wedge F_{k,l} | \neg F_{i,j}) \rightarrow \frac{P(T_{k,l}) P(F_{k,l} | T_{k,l}) P(\neg F_{i,j} | T_{k,l} \wedge F_{k,l})}{P(\neg F_{i,j})}$

However, we know that $P(\neg F_{i,j} | T_{k,l} \wedge F_{k,l})$ is necessarily equal to 1, since the false positive rate is equal to 0.

Thus, we get $\frac{P(T_{k,l}) P(F_{k,l} | T_{k,l} * 1}{P(\neg F_{i,j})}$

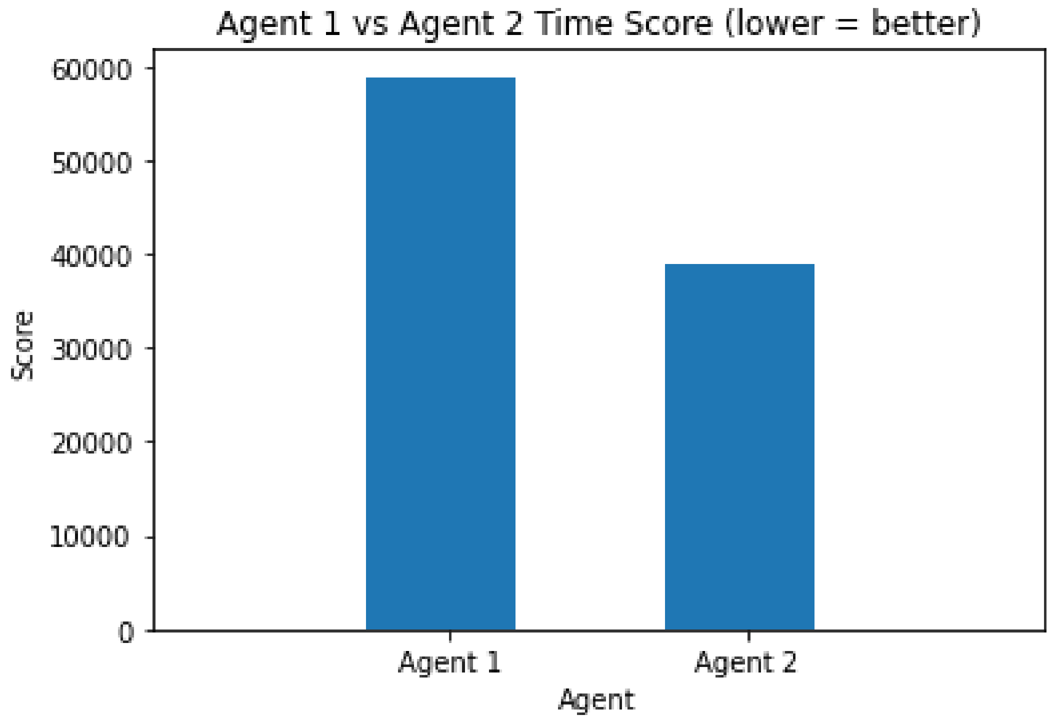
This is just $P(F_{k,l} | T_{k,l}) * \frac{P(T_{k,l})}{P(\neg F_{i,j})} = P(F_{k,l} | T_{k,l}) * P(T_{k,l})_{t+1}$

We already showed how to compute $P(T_{k,l})_{t+1}$ and $P(F_{k,l} | T_{k,l})$ is a given probability.

As a result, $P(T_{k,l} \wedge F_{k,l} | \neg F_{i,j})$ is easy to compute using the given values.

Problem 3: Which Basic Agent, 1 or 2, is better, on average?

After generating 10 maps and playing through each map 10 times with a random target location and initial agent location each time for each agent, Agent 2 had a lower score compared to Agent 1, as seen in the below graph of **Agent 1 vs Agent 2 Time Score** and is, therefore, better.



Problem 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?

Our improved agent in `agent3.py` looks ahead by one step in the following way:

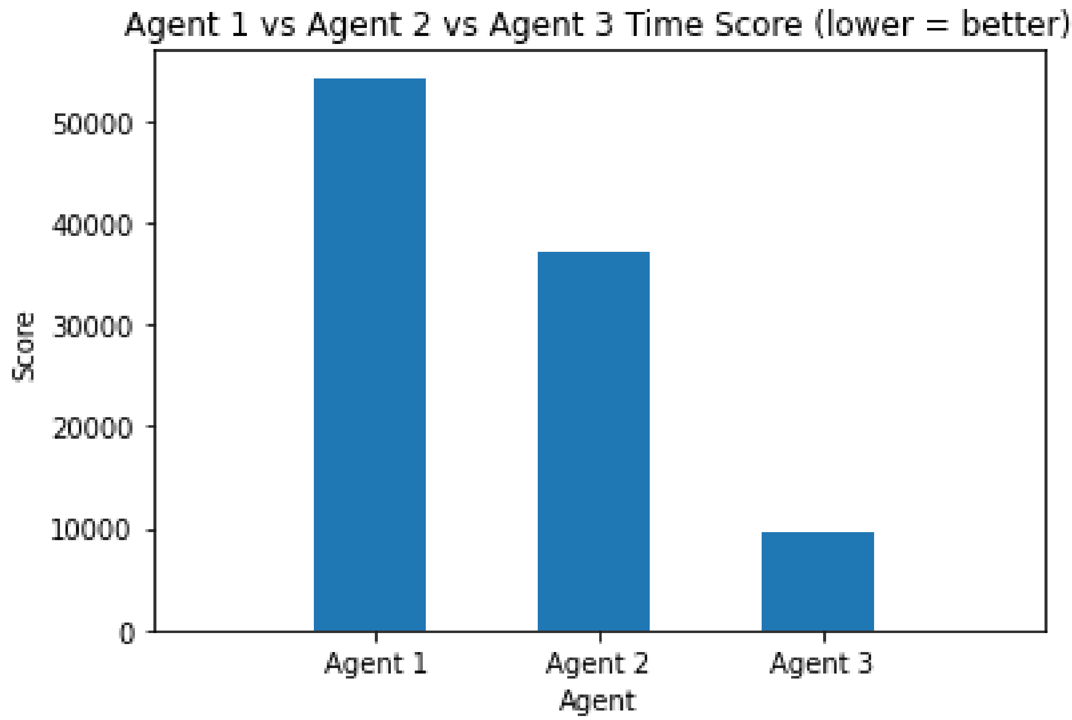
> It finds the cell with the max probability, m_t to find the target and then looks at the belief state B_{t+1}assuming we fail to find anything in cell m_t. Next, it will find the next max, m_{t+1} in B_{t+1} and compare the distances. If m_{t+1} is closer to the current cell than m_t, then we will search m_{t+1}.

In addition, this time around, the agent will search a cell multiple times before it determines the new belief matrix. The number of times an agent searches a cell is based on the false negative rate for the cell. The cell is searched until the expected value of $E(F_{i,j}|T_{i,j}) \geq 1$. Thus, we would calculate the number of times to search a cell using $\left\lceil \frac{1}{P(F_{i,j}|T_{i,j})} \right\rceil$. This is more effective than agent 1 and 2 because the improved agent will not waste time going to a further cell if it may fail and the next most effective cell to search is closer. Furthermore, multiple searches at the same cell allows for a greater change in the belief state at a much lower cost.

With more time we would design an algorithm that looks at the lowest cost of a cell. Cost is computed as:

$$\text{distance to cell i,j} + \text{ \# times cell i,j has been searched} + [1 - P(F_{i,j} \wedge T_{i,j})] * (\text{cost of next max cell})$$

As we realize this cost function can be infinitely recursive. Thus, to preserve time, we would implement a threshold to avoid infinite calculations. In `agent4.py`, we limited it by doing only one step look ahead, unfortunately due to time constraints we could not test this algorithm. This algorithm would be the best since it accounts for a failure for the searching the max cell by discounting future costs instead of just picking a cell based on highest probability. This gives a much more information about the cost of searching the cell compared to the other cells.



Bonus: Clever Acronym

Our acronym for the advanced agent algorithm is Future State Multi Search (FSMS).

Bonus: LaTeXed Report

This report was written and compiled using LaTeX.