CS 520 - Bayesian Search

Jonathan King, Alfonso Buono & Kuber Sethi

CS 520: Intro to Artificial Intelligence

1 Question 1

We can use Bayes' theorem to efficiently update the belief state given observations up until time t. In other words, we can use Bayes' theorem to calculate:

$$P(Target\ in\ Cell_i \mid Observations_t \ \wedge \ Target\ not\ found\ in\ Cell_i)$$

Since we are making several observations throughout the process of our search, we can continually update the probability of the target being in $Cell_j$ using the following formulas.

$$p = P(Target\ in\ Cell_i)$$

and

$$p' = P(Target\ in\ Cell_i \mid Target\ not\ found\ in\ Cell_i)$$

where p' is the updated probability of the target being present in $Cell_i$ after observing a failure in $Cell_j$. Note that i, j may be equal. Let us also note that:

$$\begin{aligned} q &= P(Target\ found\ in\ Cell_i \mid Target\ is\ in\ Cell_i) \\ &= 1 - P(Target\ not\ found\ in\ Cell_i \mid Target\ is\ in\ Cell_i) \end{aligned}$$

We have two cases to consider. $Cell_i = Cell_j$ and $Cell_i ! = Cell_j$

Case $Cell_i = Cell_j$:

$$P(Target\ in\ Cell_i \mid Target\ not\ found\ in\ Cell_i)$$

$$=$$

$$\frac{P(Target\ not\ found\ in\ Cell_i \mid Target\ in\ Cell_i) * P(Target\ in\ Cell_i)}{P(Target\ not\ found\ in\ Cell_i)}$$

$$=$$

$$\frac{P(Target\ not\ found\ in\ Cell_i \mid Target\ in\ Cell_i) * P(Target\ in\ Cell_i)}{1 - P(Target\ found\ in\ Cell_i)}$$

$$=$$

$$\frac{P(Target\ not\ found\ in\ Cell_i \mid Target\ in\ Cell_i) * P(Target\ in\ Cell_i)}{1 - P(Target\ found\ in\ Cell_i \mid Target\ in\ Cell_i) * P(Target\ in\ Cell_i)}$$

Therefore,

$$P(Target\ in\ Cell_i |\ Target\ not\ found\ in\ Cell_i)$$

$$=$$

$$\frac{(1-q)*p}{1-(q*p)}$$

Case $Cell_i != Cell_j$:
Let $r = P(Target\ in\ Cell_j)$

$$P(Target\ in\ Cell_j |\ Target\ not\ found\ in\ Cell_i)$$

$$=$$

$$\frac{P(Target\ not\ found\ in\ Cell_i \mid Target\ in\ Cell_j) * P(Target\ in\ Cell_j)}{P(Target\ not\ found\ in\ Cell_i)}$$

$$=$$

$$\frac{P(Target\ not\ found\ in\ Cell_i \mid Target\ in\ Cell_j) * P(Target\ in\ Cell_j)}{1 - P(Target\ found\ in\ Cell_i)}$$

$$=$$

$$\frac{P(Target\ in\ Cell_j)}{1 - P(Target\ found\ in\ Cell_i |\ Target\ in\ Cell_i) * P(Target\ in\ Cell_i)}$$

Therefore,

$$P(Target\ in\ Cell_j |\ Target\ not\ found\ in\ Cell_i)$$

$$=$$

$$\frac{r}{1-(q*p)}$$

where $r = P(Target\ in\ Cell_j)$

2 Question 2

Once we have calculated the probability of a target being present in a given cell using the previously defined equations, we can easily find the probability of a target being found in the cell. The probability of a cell being found in a cell is the following:

$$P(Target\ found\ in\ Cell_i)$$

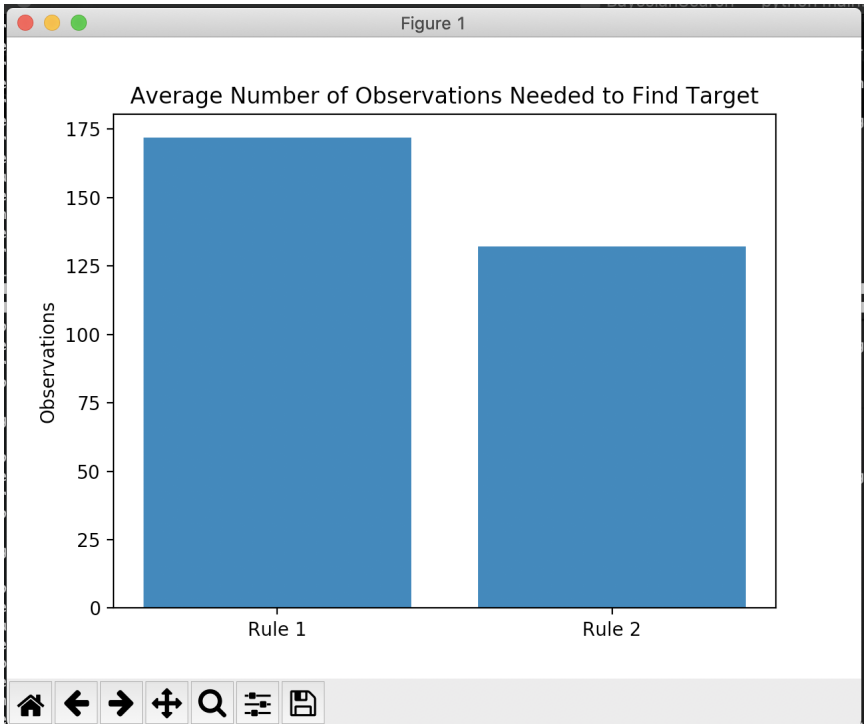
$$=$$

$$1 - P(Target\ not\ found\ in\ Cell_i \mid Target\ in\ Cell_i) * P(Target\ in\ Cell_i)$$

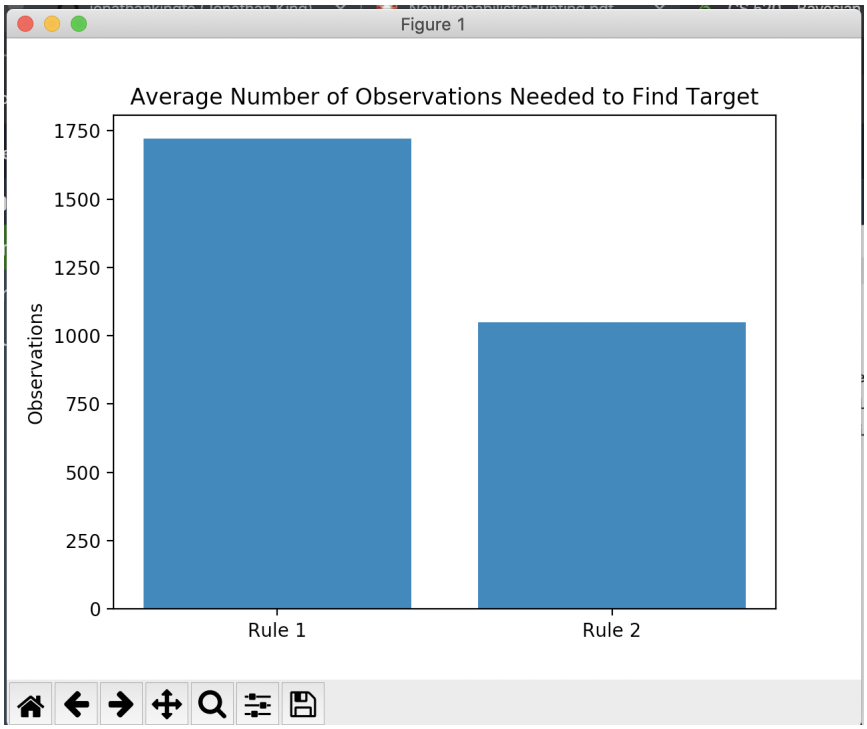
3 Question 3

We implemented the rules in such a way that informs us where we should select our next query location. Based on the rule given, we will either return the

location with the highest probability of finding the target or the location with the highest probability of containing the target. Based on a fixed map size, we found the average number of searches needed to find the target for each rule. This yielded the following.



Fixed Map Size of 20



Fixed Map Size of 50

In a fixed map, we determined that Rule 2 performs better, and that this holds across multiple maps. This is due to the fact that Rule 2 takes into account the landscape of other cells, which leads to a more accurate probability. This also makes sense intuitively. In most cases, it is better to search the locations with the highest probability of finding the target.

4 Question 4

For this section, we used the same algorithm to find the most optimal place to search as in the previous question. However, in order to account for the travel cost of moving around the grid, we added a heuristic to find the most optimal spot.

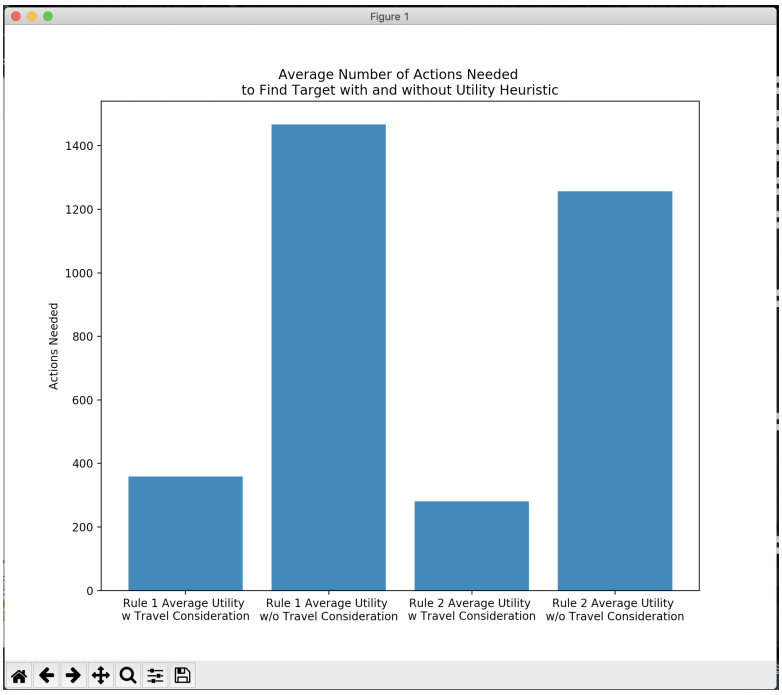
The heuristic we used was the Manhattan distance between the current location and the potential query location. Instead of finding the maximum probability of finding/containing the target, we found the maximum of:

$$\frac{P(Target\ in\ Cell_i)}{1 + Manhattan\ Distance\ To(Cell_i)}$$

or

$$\frac{P(Target\ found\ in\ Cell_i)}{1 + Manhattan\ Distance\ To(Cell_i)}$$

depending on the rule.



This data makes sense with our intuition. When we use the heuristic to decide the most optimal search point, we are able to find the target in less actions than when we don't use it. This is because we are minimizing the travel cost along with maximizing the probability based on our given rule. We also see that Rule 2 still outperforms Rule 1, as seen in our previous response.

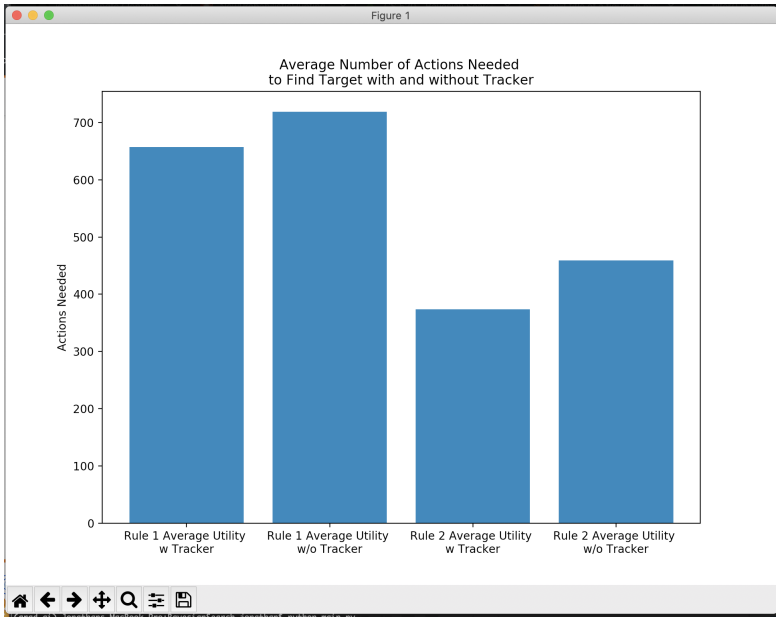
5 Question 5

After working on the project and applying the different rules, we were able to recognize a connection between the project, our results, and the joke that was given. In the joke the drunk man is searching for his keys in the light. Despite his keys not being there, since he says that they are in the park, he continues to look around by the streetlight. These actions are justified in that the agent, or drunk man, have a higher chance of locating the target in a spot that has a higher rate of finding it. However, just because there is a high probability of finding it does not mean that it is the best place to look in all cases. If the target were in a "caves" or "forest" tile, the find_probability will drop significantly because of the high rate of a false negative. Therefore, the agent is going to typically search "flat" or "hilly" tiles just as the drunk man is searching under

the streetlight.

6 Question 6

In order to utilize the additional information given by the tracker, we only had to add an extra check within our select query function. When using the tracker, we would filter out the terrain type that was returned by the tracker. This would restrict the agent to look only at cells in which the target could be.



This plot matches with what we expected. Again, Rule 2 was faster than Rule 1 both when using and when not using the tracker information. In both Rules, the tracker information decreased the amount of actions needed in order to find the target. This is because the tracker information limited the number of cells in which our agent decided to query from. Therefore, the probability of the cell being present in our query location would be higher than the probability of it being present in a query location in which we included cells where we know the target could not be.