Assignment 3

Farhat Alam, Karthik Paka, Matan Shachnai, Andrea Wu
198:520 Intro to Artificial Intelligence
Rutgers University

March 27, 2019

1 Stationary Target

1) We are interested in updating our belief state based on past observations and failed searches. Following the Markovian assumption, we are specifically interested in the very last observation (and failed search) made prior to our current search as it encompasses past observations. In order to illustrate this idea, we will use a small example of search and destroy and generalize.

For a $10X10$ board (dim = 10), our initial belief ($t = 0$) of where the target can be found is represented by

$$\text{Belief}_0[\text{cell}_\text{i}]_= \frac{1}{\#\text{ of cells}} \rightarrow \text{Belief}_0[\text{cell}_\text{i}] = \frac{1}{100}.$$

We also know that for each type of land associated with a given cell, there is a given probability of a false negative (complements added for general reference):

$$P(\text{Target not found in cell}_i|\text{Target in cell}_i) = \begin{cases} 0.1 & \text{cell}_i \text{ is flat } \complement = 0.9 \\ 0.3 & \text{cell}_i \text{ is hilly } \complement = 0.7 \\ 0.7 & \text{cell}_i \text{ is forested } \complement = 0.3 \\ 0.9 & \text{cell}_i \text{ is maze of caves } \complement = 0.1 \end{cases}$$

Given this information, we search a random cell. Let's assume we do not find the target in this cell, but we do find out the cell is forested. At this point, we have valuable information that we can use to update our current state of belief about the whereabouts of the target. Specifically, we know that

$$P(\text{Target not found in cell}_i|\text{Target in cell}_i) = 0.7$$

and that

$$P(\text{Target in cell}_i) = 0.01.$$

To find the new probability of cell$_i$, after the failed search, we want to find the probability of P(Target not found in cell$_i$ $\wedge$ Target in cell$_i$). We can derive this based on our knowledge so far. Using Bayes' theorem, we get the following:

$$P(\text{Target not found in cell}_i|\text{Target in cell}_i) = \frac{P(\text{Target not found in cell}_i \wedge \text{Target in cell}_i)}{P(\text{Target in cell}_\text{i})}$$

In our example, we get that

$$0.7 = \frac{P(\text{Target not found in cell}_i \wedge \text{Target in cell}_i)}{0.01}$$

$$0.007 = P(\text{Target not found in cell}_i \wedge \text{Target in cell}_i)$$

This can also be expressed as P(Target in cell$_i$)P(Target not found in cell$_i$|Target in cell$_i$) = (0.01)(0.7) = 0.007.
Given this new information, we need to update the probability of the target being in any other cell on the grid to reflect this change in belief without changing the ratios of probabilities. We can do this by normalizing the whole grid to 1. Thus, we divide (or multiply by reciprocal) each cell on the grid by the sum of all current probabilities of the target being in each cell:

$$\text{For all cells in grid. cell probability} = P(\text{Target in cell}) \cdot \frac{1}{\sum_{i=0}^{dim} \sum_{j=0}^{dim} P(\text{Target in cell}_\text{ij})}$$

In our example, the summation is equal to $\frac{99}{100} + \frac{7}{1000} = \frac{997}{1000}$. The first element describes the initial probabilities of 99 other cells we have not searched, and the second element describes the cell we searched but failed to find the target. Using our above formula to normalize an individual cell (one we haven't searched), we get

$$\text{cell probability} = \frac{1}{100} \cdot \frac{1000}{997} = \frac{10}{997}$$

This is the new probability of every cell we haven't searched in this state (99 cells in this case). The last cell we need to normalize is the failed search cell which computes to

$$\text{searched cell probability} = \frac{7}{1000} \cdot \frac{1000}{997} = \frac{7}{997}$$

Now we have a new belief state using Bayes' theorem and normalization.

At this point we can transition to the next state and repeat this process based on the observations and failure made at time $t-1$ (the previous state). As the model keep updating, cells with higher initial probabilities will increase while cells with lower initial probabilities will decrease, thus we will eventually get closer to finding the target.

2) Our calculations from question (1) compute the probability that a given cell contains the target in it given prior observations and search failures. In this case we are interested in the probability of finding the target given that it is in the cell we are searching. We are now looking at the complement of the false negative of each cell type (values given in question 1) and we get that the probability we are looking for can be expressed in the following way:

$$P(\text{Target in cell}_\text{i})P(\text{Target found in cell}_i|\text{Target in cell}_i)$$

This formula is the initial search we conduct at $t = 0$, which is just equal to $\frac{1}{\text{\# of cells}}$. It is important to note that while cells may have the same probability of containing the target, it is much more likely that we will find the target in a Flat land rather than in a Cave; finding the target is much depended on the type of land of a given cell. At every state after our initial belief state, we need to take into account the prior observations made (at time $t - 1$) to calculate the correct probability of finding the target in a given cell. This can be expressed as:

$$P(\text{Target in cell}_\text{i}|\text{observations}_\text{t-1})P(\text{Target found in cell}_i|\text{Target in cell}_i)$$

3) Rule 1 corresponds to our solution to (1); we find $max(P(\text{target in cell}_i))$ based on observations made up to time t (see solution to 1). Rule two signifies the probability of finding a target given it is in a cell - this corresponds to our solution to (2) and the foundations set by (1), namely, ascertaining the probablity of a given cell containing the target. We find $max(P(\text{target in cell}_i) \cdot P(\text{found in cell}_i|\text{target in cell}_i))$. This varies based on land type as we multiply P(target in cell$_i$) by the complement of each land type rate of false negative (see 1).
In order to compare the effectiveness of each rule, for a 50 by 50 grid, we simulated 50 trials where the map remains the same and 50 more trials where a completely new map is generated:

50 trials: 50X50 grid, same map (target position does not change):
avg search (rule 1) : 5416
avg search (rule 2) : 5039.06

50 trials: 50X50 grid, same map (target position changes each round):
avg search (rule 1) : 5629.8
avg search (rule 2) : 4055.12

50 trials: 50X50 grid, different maps:
avg search (rule 1) : 5911.4
avg search (rule 2) : 5198.14

These results suggest that searching according to rule 2, highest probability of finding target, is most effective given this grid setup. These results are not surprising because rule 2 is biased towards searching easy land types such as Flat and Hill land types. Since our initial probabilities are such that half of the grid is made up of these land types, it only makes sense that rule 2 surpasses rule 1. These results would not hold if the land type distribution and ease of finding a target were different.

To further this claim, we swapped the land type distribution of each cell to the following: 0.1 for Flat, 0.2 for Hill, 0.4 for Forest, and 0.3 for Cave. The results are as follows:

50 trials: 50X50 grid, same map (target position does not change):
avg search (rule 1) : 7352.4
avg search (rule 2) : 10694.8

In the previous cases, we saw that rule 2 was always more efficient in finding the target. In this land type distribution, however, rule 1 is better. This is because there are more difficult terrains which rule 2 tends to avoid. Therefore, rule 1, a uniform search algorithm across all cells regardless of land type, is better in this case.

4) The approach we chose involved examining the current cell along with its neighbors above, below, on the right, and on the left and determining which cell among the 5 had the highest probability for each rule. If we searched the current cell, we would add one action to our total number of actions, and if we moved to a neighboring cell and searched it, we would add two actions to our total number of actions. Being limited in movement, we needed to find an approach that leveraged the information that we had available to us while also minimizing the total number of actions. By always choosing the cell with the maximum probability available, we maximized our chances of finding the target cell during any given step. The results are as follows:

50 trials: 50X50 grid, same map (initial target position does not change):
avg actions (rule 1) : 14703.22
avg actions (rule 2) : 40593.5

50 trials: 50X50 grid, different maps:
avg actions (rule 1) : 12296.78
avg actions (rule 2) : 20251.28

We found that for both cases, rule 1 required fewer searches than rule 2, which is not unreasonable given the movement restriction. When comparing the performance to that of the previous question, it is important to consider that here we look at the average number of actions (both moving and searching), while in the previous question we simply look at the average number of searches. Taking this into account, we see that this approach requires more searches, on average, when compared to the previous question. This is consistent with our expectations because we are not necessarily able to search the cell with the highest probability at any given point due to our restricted movement. As a result, we choose the best option from those available to us.

5) In light of the results of this project, we find that the joke has valid reasoning behind it. The drunk's strategy to search under a street light even though he lost his keys in the park, initially seemed like a waste of effort, but upon further thought, we can see that it can be a good plan. The drunk's idea correlates to Rule 2 of the project - where you check cells with the greatest probability of finding the target if it actually is in that cell. The results of our experiments show that this method, on average, requires searching fewer cells in order to find the target (as described by question 3).

In the context of the drunk man searching for his keys this methodology works because he is essentially eliminating cells from his search problem by checking ones that he knows will immediately lead to a quick yes or no answer. If he instead went searching in the park first, he would never know how after how many rounds of searching he should stop looking at an area, because he can't be sure if he didn't find the keys in one spot because they actually aren't there or just because he couldn't see well enough. So after every round of searches, he still hasn't been able to eliminate much area from his search space, so he will have more to check in the next round than if he had started with places that he could quickly eliminate or quickly find the key in. To put it in simple terms, even if you believe that the key is most likely in the park, it doesn't waste much time to first do a quick search under the street lamps just to make sure you didn't drop your keys there. It would be inefficient to search only under the lamps for the entire night. Conversely, if you start in the park, it will take you a while to get any solid results that you know for sure, and it will be nagging you in the back of your mind that maybe by some fluke you dropped your keys outside the park under the street lamp, so you never really get conclusive results and you get frustrated searching areas over and over again. Using rule 2 for searching, as the drunken man did, is supported by our solution to question (3).

2 Moving Target

1) The belief state at time 0 is given by $\frac{1}{\text{\# of cells}}$; every cell has the same probability of containing the target initially. The main difference in the moving target scenario is that we are given valuable information once we get the land types in which the target moved between. In this scenario once we are given the two target land types, we can effectively rule out any cell that is not one of the two target land types, meaning that their probability of containing the target is 0. Before we set all non land type cells probability to 0, we sum all of their probabilities and distribute them evenly between target cells. After doing so we normalize the probabilities to 1. Thus we continue updating and normalizing cells until we reach the target.

In order to compare the effectiveness of each rule, for a 50 by 50 grid, we simulated 50 trials where the map remains the same and 50 more trials where a completely new map is generated.

50 trials: 50X50 grid, same map (initial target position does not change):
avg search (rule 1) : 3238.68
avg search (rule 2) : 4038.14

50 trials: 50X50 grid, same map (initial target position changes each round):
avg search (rule 1) : 1982.72
avg search (rule 2) : 2800.74

50 trials: 50X50 grid, different maps:
avg search (rule 1) : 3100.12

avg search (rule 2) : 3217.32

The first thing we notice is that the number of searches required to find the target is significantly less (roughly 50% less searches required) than when the target is stationary. As the search progresses, the seeker gets information about the movement pattern of the target. This helps reduce the number of searches required to find the target as can been seen by the search results above. These results also suggest that searching according to rule 1, highest probability of cell containing target, is most effective given this grid setup. These results make sense because the target is no longer stationary and is actively moving through the grid. This means that the target wanders into all land types, thus the search algorithm that would fit best in this scenario is a one that searches breadth rather than depth. Rule 1 is unbiased towards ease of finding target in a given cell and thus searches across the board uniformly where rule 2 tends to search the easiest cells first, meaning that it would potentially miss the target because it would rather search a flat land rather than a cave.

2) Results of simulation:

50 trials: 50X50 grid, same map (initial target position does not change):
avg actions (rule 1) : 18835.3
avg actions (rule 2) : 20858.94

50 trials: 50X50 grid, same map (initial target position changes each round):
avg actions (rule 1) : 14041.7
avg actions (rule 2) : 20343.92

50 trials: 50X50 grid, different maps:
avg actions (rule 1) : 31062.38
avg actions (rule 2) : 24510.44

We found that rule 1 required less actions on average when considering the same map, which is consistent with our results in 1.4. However, for the case where we average the results of searching different maps, we find that rule 2 results in fewer actions. We attribute this difference to the difference in difficulty of finding the target for different maps, as each different map has a different initial target position, terrain, and search starting position, which has a direct impact on the results. As such, some maps may require far more searches than others, skewing the average in that direction. We recognize that using a larger number of trials to compute the average would result in a better mean, but this approach is illogical given the magnitude of the computation time. As a result, we find 50 trials to be a reasonable number of trials given this restriction.