Fall 2018 CS 520: #3 Wes Cowan

Due by Nov. 18 11:55 pm.

GroupMember 1: Lemin Wu ID: 171008940, NetID: lw514
GroupMember 2: Guo Chen ID: 173006648, NetID: gc538
GroupMember 3: Yujie Ren ID: 167006818, NetID: yr181

1. A Stationary Target

1) Given observations up to time *t* (Observations *t*), and a failure searching Cell *j* (Observations *t + 1* = Observations *t* ∧ Failure in Cell *j*), how can Bayes' theorem be used to efficiently update the belief state, i.e., compute:

P (Target in Cell *i* | Observations *t* ∧ Failure in Cell *j*). (4)

Solution:

Suppose I_i be the event that the target is in cell *i* and F_j be the event that the target can be found in Cell *j*. Also let O_t denote the observations at time *t*. Based on information we obtained from the statements, we have

$P(I_i \mid O_{t+1}) = P(I_i, O_{t+1}) / P(O_{t+1}) = P(O_t)P(I_i \mid O_t)P(O_{t+1} \mid I_i, O_t) / P(O_{t+1})$.

Intuitively, the observations at time *t* is definitely in certain, therefore $P(O_t) = 1$. In addition, based on the knowledge we have from the given statements, we can conclude that $P(O_{t+1}) = \sum_k P(O_{t+1} \mid I_k)\, P(I_k)$. Since the observations at time *t+1* means a union of the observations at time *t* and an obtained fact that we fail to find the target in cell *j* at that moment, now the equation can be transformed as $P(O_{t+1}) = \sum_k P(O_{t+1} \mid I_k)\, P(I_k) = \sum_k P(O_t, \overline{F_j} \mid I_k)\, P(I_k) = \sum_k P(\overline{F_j} \mid I_k)\, P(I_k)$. For simplicity of resolving the probability equation, we assume $\alpha = 1 / P(O_{t+1}) = 1 / \sum_k P(\overline{F_j} \mid I_k)\, P(I_k)$. Therefore,

$P(I_i \mid O_{t+1}) = P(I_i \mid O_t)P(O_{t+1} \mid I_i, O_t) / P(O_t) = \alpha \bullet P(I_i \mid O_t)P(O_{t+1} \mid I_i, O_t) = \alpha \bullet P(I_i \mid O_t)P(O_t, \overline{F_j} \mid I_i, O_t) = \alpha \bullet P(I_i \mid O_t)P(\overline{F_j} \mid I_i)$.

For cell i and j, if i equals j, $P(I_i \mid O_{t+1}) = \alpha \bullet P(I_i \mid O_t)P(\overline{F_i} \mid I_i)$, otherwise $P(I_i \mid O_{t+1}) = \alpha \bullet P(I_i \mid O_t)$ since $P(\overline{F_j} \mid I_i)$ always equals to 1 because target should be either in *i* or in *j* exclusively.

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:

P (Target found in Cell i | Observations t)? (5)

Solution:

Suppose I_i be the event that the target is in cell i and F_j be the event that the target can be found in Cell j. Also let O_t denote the observations at time t. And given information we obtained from question 1) and the given statements,

$$P(F_j \mid O_t) = P(F_j \mid I_i) P(I_i \mid O_t) = (1 - P(\overline{F_j} \mid I_i)) P(I_i \mid O_t).$$

3) Consider comparing the following two decision rules:

– 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.

For either rule, in the case of ties between cells, consider breaking ties arbitrarily. How can these rules be interpreted / implemented in terms of the known probabilities and belief states? For a fixed map, consider repeatedly using each rule to locate the target (replacing the target at a new, uniformly chosen location each time it is discovered). On average, which performs better (i.e., requires less searches), Rule 1 or Rule 2? Why do you think that is? Does that hold across multiple maps?

Solution:

As we have seen previously, both rules can be presented in a way as shown in the the questions 1) and 2), using known probabilities and belief states.

And to demonstrate how well both rules perform, we carried out a series of experiments by making both rules separately perform 1000 times for randomly generated maps with size 50 * 50. Besides, for every each 1000 repetitions, we make the target be in a cell with one of the four types of landscapes. And for convenience of carrying out experiments, we always set a cell with flat landscape as the one containing the target so that it won't take too much time trying to find out the target, because is mostly likely to find a target in a flat typed cell compared with other types. The average steps to successfully find a target in terms of different policies are shown below (for a better comparison of both rules, we set the third rule of randomly picking up a cell to search each time). The first row indicates different types of the cell containing the target.

	Flat	Hilly	Forested	Maze
Random Search	2829.2	3449.2	8291.7	26839.5

Rule 1	3089.0	4923.2	8311.4	8648.8
Rule 2	975.1	2608.2	5665.3	12485.8

Based on data we obtained from experiments, it is obviously to conclude that Rule 2 requires less steps to find the target than Rule 1 except in the last case. One possible reason for that is that Rule 2 takes into account of probability of finding the target, which varies for different landscape types and thus cannot be ignored. And from the experiment we have seen, such a conclusion holds across multiple maps with variant sizes.

4) Consider modifying the problem in the following way: at any time, you may only search the cell at your current location, or move to a neighboring cell (up/down, left/right). Search or motion each constitute a single 'action'. In this case, the 'best' cell to search by the previous rules may be out of reach, and require travel. One possibility is to simply move to the cell indicated by the previous rules and search it, but this may incur a large cost in terms of required travel. 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? Derive a decision rule based on the current belief state and current location, and compare its performance to the rule of simply always traveling to the next cell indicated by **Rule 1** or **Rule 2**. Discuss.

Solution:
The belief board is maintained as same as the above part: to update Belief[Celli] = α*Belief[Celli]*P(- found | target in Celli) after we dig Cell i and don't find the target. Then an instant belief board will be calculated based on the belief board in each step: Instant_Belief[Celli] = Belief[Celli]/ Manhattan distance (from current cell to the highest probability belief cell in the board). Since we can only move one step in this case, we choose the neighbor cell that is closest to the cell with highest probability in the instant belief board as our next cell to dig. By using Instant_Belief[Celli] as the value used to make decisions, the distance factor is included in the system: the further the cell locates, the lower its value is.

Similar to question 3), we carried out 1000 experiments, averaged the result steps on maps of 50 * 50 and recorded below, except that we no longer make comparison with random case:

	Flat	Hilly	Forested	Maze

Rule 1	2713.4	5370.0	8187.7	9693.4
Rule 2	2192.9	3176.1	7713.8	23970.7

5) An old joke goes something like the following:

A policeman sees a drunk man searching for something under a streetlight and asks what the drunk has lost. He says he lost his keys and they both look under the streetlight together. After a few minutes the policeman asks if he is sure he lost them here, and the drunk replies, no, and that he lost them in the park. The policeman asks why he is searching here, and the drunk replies, "the light is better here".

In light of the results of this project, discuss.

Solution:

This joke is very similar as the stationary target part in this project: firstly, the target's (key) location is unknown. In addition, there is chance of not finding the target though the target is there. The probability depends on the place (p(not found | terrain) in the project and lightness in the joke). Rule 1 is to search the place with the highest probability of containing the target. Rule 2 is to search the place with the highest probability of finding the target. Since P(finding the target) = P(containing the target)*P(finding the target | containing the target), it actually makes sense for the drunk man to search the lighter place first though it's still ridiculous to only consider the lightness. From the result we observed in the project, Rule 2 works generally better than Rule 1. So for the drunk man, the better strategy of searching his key is to consider together the probability of where the key is and the lightness of the place.

2. A Moving Target

1) Implement this functionality in your code. How can you update your search to make use of this extra information? How does your belief state change with these additional observations? Update your search accordingly, and again compare **Rule 1** and **Rule 2**.

Solution:

The source code of this part is attached with this report.

For a different situation where the target is moving as far as it is not found, we need to make use of the type report as a new evidence for updating. However, we need to notice one important difference here. In the situation of moving target, since a type report would be generated as far as the target is not found, it indicates which type(s) should be noticed and thus all cells with other types definitely contain no target at all. In this case, the belief states of these cells should be zero. In the next moment, suppose the target is not found and moves to a cell with the third type. In this case, if we still use the belief states in the last moment, it is clearly that such belief states obtained last moment would ‘propagate’ to its surroundings. After several unsuccessful searches or target movements, all the belief states would become zero, which is quite opposite to our expectations.

In order to resolve this problem, we argue that the belief distribution in each moment when a target moves would be totally different from last moment. In other words, we can regard all the belief states to be the initial one and adjust them merely based on the target movement at the exact moment, which has nothing to do with the position of the target in the last moment. By using this method, we successfully solved the problem and made belief states of all cells change in a reasonable way.

Just similar in question 3) in section 1, we carried out a series of experiments and here is data we obtained as shown below. The first row indicates different types of the cell containing the target.

	Flat	Hilly	Forested	Maze
Random Search	148.4	142.7	102.5	139.5
Rule 1	54.8	71.9	22.5	39.0
Rule 2	22.7	20.2	10.1	18.3

2) Re-do question 4) above in this new environment with the moving target and extra information.

Solution:

It’s kind of a combination of the above: instead of making decision on changing_landscape_belief, the next cell to dig is based on changing_landscape_belief/ Manhattan distance (from current cell to the highest probability belief cell in the board). Similarly since we can only move one step in this case, we choose the neighbor cell that is closest to the cell with highest probability in the instant belief board as our next cell to dig.

Similar to question 4) in section 1, we carried out 1000 experiments, averaged the result steps on maps of 50 * 50 and recorded below:

	Flat	Hilly	Forested	Maze
Rule 1	107.9	49.4	104.8	109.9
Rule 2	58.3	49.4	69.4	54.3