Suyu Huang, Cong Deng, Bowei Feng

CS520 INTRO TO AI

17 November 2018

Probabilistic Search (and Destroy)

1. A Stationary Target

1)Given observations up to time t ($Observations_t$), and a failure searching $Cell_i$ ($Observations_{t+1} = Observations_t \wedge Failure\ in\ Cell_i$), how can Bayes' theorem be used to efficiently update the belief state, i.e., compute:

$$P(Target\ in\ Cell_i \mid Observations_t \wedge Failure\ in\ Cell_i)\ ?$$

Ans: Our initial knowledge about the landscape is that every cells' probability of containing the target is 1/(size of landscape). Every time we search the cell with maximum hope to contain the target by doing the following if-else statement:

IF we find the target, then we return the steps.

Else we update the belief of remaining cells.

We know that:

Belief[Cell i] = P (Target in Cell i | Observations Failure).

Using Bayesian Theorem:

P (Target in Cell i | Observations Failure)

= P (Observations Failure | Target in Cell i) * P (Target in Cell i) / P(Observations Failure).

For P(Observations Failure):

P(Observations Failure)

= P(Observations Failure ^ Target in Cell) + P(Observations Failure ^ Target not in Cell)

= P(Observations Failure | Target in Cell)*P(Target in Cell) + P(Observations Failure | Target not in Cell) * P(Target not in Cell)

Based on the formula, we can calculate the Belief for remaining cells.

Belief[Cell i] = Belief[Cell i] + Belief[Cell i] * delt(Belief[Cell j]) / (1 - Belief[Cell j])

Then we can solve the $P(Target\ in\ Cell_i \mid Observations_t \wedge Failure\ in\ Cell_i)$

P(Target in Cell i | Observations t ^ Failure in Cell i)

= P(Target in Cell i ^ Observations t ^ Failure in Cell i) / P(Observations t ^ Failure in Cell i)

=P(Observations t)*P(Target in Cell i | Observations t)*P(Failure in Cell i | Target in Cell i ^ Observations t) / P(Observations t ^ Failure in Cell i)

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:

$$P(Target found\ in\ Cell_i \mid Observations_t)\ \textbf{?}$$

Ans:The probability of finding a target in a cell is,

$$P(Target found\ in\ Cell \mid Target\ in\ cell) * P(Target\ in\ Cell \mid Observations)$$

And we know that $P(Target\,found\ in\ Cell \mid Target\ in\ Cell)$ is 0.9, 0.7, 0.3, 0.1 for flat, hilly, forested and a maze of caves.

And as we said in question 1, we could get the value of $P(Target\ in\ Cell \mid Observations)$ of each cell. At first time, when t=0, the probability is 1/2500 for each cell.

So, we can finally get that, $P(Target\,found\ in\ Cell_{\,i} \mid Observations_t)$ equal to

$$P(Target\,found\ in\ Cell_{\,i} \mid Target\ in\ Cell_{\,i}) * P(Target\ in\ Cell_{\,i} \mid Observations_{\,t-1})$$

And $P(Target\ in\ Cell_{\,i} \mid Observation_{\,0})$ is 1/2500.

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?

Ans:

- Rule 1

This rule can be implemented by using the belief of each cell. For time t, we search the cell with the maximum belief.

- Rule 2

  This rule can be implemented by using the equation in question 2. The next cell we will search is based on the type of terrain. And we can easily get a search order: Flat > Hilly > Forested > Caves.

Average number of searches according to target in different terrain

Flat		**Hilly**		**Forested**		**Caves**	
Rule 1	Rule 2	Rule 1	Rule 2	Rule 1	Rule 2	Rule 1	Rule 2
1223	281	1335	848	1374	1609	1392	5267

From the table above, we could get that,

- Rule 2 performs better than Rule 1 in Flat and Hilly.

  In Flat, Rule 1 needs 334% more searches than Rule 2, since there are 0 to 20 percent of cells are flat, Rule 2 starts from flat is more efficient than Rule 1 which searches all cells.

  In Hilly, Rule 1 needs 36.4% more searches than Rule 2, which is smaller compared to flat since Rule 2 searches 20 to 50 percent of cells(flat plus hilly).

- Rule 1 performs better than Rule 2 in Forested and Caves.

  In Forested, Rule 2 needs 17.4% more searches than Rule 1, since now, Rule 2 would search 50 to 80 percent of cells.

In Caves, Rule 2 needs 278.3% more searches than Rule 1, since Rule 2 needs to explore 80 to 100 percent of cells.

And then, we should see all possible position for target by using expected number of searches

Rule 1:

$0.2 * 1223 + 0.3 * 1335 + 0.3 * 1374 + 0.2 * 1392 = 1335.7$

Rule 2:

$0.2 * 281 + 0.3 * 848 + 0.3 * 1609 + 0.2 * 5267 = 1846.7$

We can see that, totally, Rule 1 performs better than Rule 2.

And then, this result does not hold for multiple maps since the number of searches is based on the distribution of terrain type.

We create a new map here, in which all cells are Forested. We could find that Rule 1 and Rule 2 perform equally for this new map. However for the original map, Rule 1 performs better than Rule 2 here since number of forested cells is 50 to 80 percent for all cells.

So we can see, the distribution of terrain type influences the performance of two rules, we can not give a proper comparison of two rules if we have no knowledge about maps.

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.

Ans:As for this specific situation, we need to use fixed distance to update the belief state. The fixed distance is showed below:

$$cost = ManhattDistance\, from\ current\ to\ Cell\ with\ Maximum\ Hope$$

Note that: Cell with Maximum Hope is defined based on rule 1 or rule 2.

For each rule we have:

$$rule\ 1 : new\ probability\ state = belief\ state\ /\ (offset\ +\ cost)$$

$$rule\ 2 : new\ probability\ state = prob\, find\ state\ /\ (offset\ + cost)$$

We will choose the cell with maximum value in new belief state as the next cell to be searched.

Flat		**Hilly**		**Forested**		**Caves**	
Rule 1	Rule 2	Rule 1	Rule 2	Rule 1	Rule 2	Rule 1	Rule 2
3922	1284	6554	1422	4887	3457	5605	7387

The find probability of terrain is positive related with the average steps of finding the targets. And average steps for flat and hilly terrains Rule 1 seems to be higher than Rule 2 but the situation is opposite in forested and caves.

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.

Ans:The drunk man behaved like what we did in question 3 with rule 2. Even though he knows that he lost his keys in the park, he still searched his keys under the streetlight, since he thinks the light is better here which means he has higher probability to find his keys here. And we know even though it has higher probability for the drunk man to find keys here, the probability of keys being here is nearly 0, so the probability of finding the keys is a little. In turn, the probability of keys being in park is much higher, but the drunk man just saw the difficult of finding the keys in park.

And now, based on the result of our project, the drunk man should follow these true probability of his keys being in the park and being under the streetlight, finally finding his keys in the park and under the street light. If now the probability of his keys being in the park is equal to being under the streetlight or is not much greater than being under the streetlight, the drunk man would better to search under the streetlight firstly. If now the drunk man is not sure where he lost his keys and the probability of successfully finding the keys under the streetlight is much higher than in the park, it is more efficient for the man to search the steerlight at first, which means he could cost less time. And if the man knows the proper value of these probability, calculates and then finds the probability of successfully finding the keys in the park and under the streetlight is almost equal, he would better search the streetlight firstly which is close to him before going to the park which is far away from him.

2. A Moving Target

In this section, the target is no longer stationary, and can move between neighboring cells. Each time you perform a search, if you fail to find the target the target will move to a neighboring cell (with uniform probability for each). However, all is not lost - whenever the target moves, surveillance reports to you that the target was seen at a *Type1 x Type2* border where *Type1* and *Type2* are the cell types the target is moving between (though it is not reported which one was the exit point and which one the entry point.

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.

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

Ans:When $t = 0$, we have all cells in the belief state as 1/2500. Each time the target moves, we can get the terrain types of the start and end cells(but don't know which one is start cell or end cell). Since the target can only move to its adjacent cell, we can use the terrain types to filter the adjacent cells sets that contains these two types and set other cells' probabilities to .

STRATEGY:

1. Cell with a type which is not type1 or type must have zero probability to contain target.

2. Cell which is not adjacent to a cell with type1 or type2 must have zero probability to contain target.

3. Cell surrounded by 4 cells which all has zero probability at previous steps must have zero probability to contain target.

4. Distribute the previous probability of cells that are determined not containing target to remaining cells that have the probability of containing target.

Moving target	Rule 1	Rule 2
Average movement	2712.82	3729.37