AI Project 3: Probabilistic Search (and Destroy)

Daniel Ying dty16 Sec: 440:06, Kyle VanWageninge kjv48 Sec: 440:05

May 24, 2021

Submitter:

Honor Code:

I abide to the rules laid in the Project 3: Probabilistic Search (and Destroy) description and I have not used anyone else's work for the project, and my work is only my own and my group's.

I acknowledge and accept the Honor Code and rules of Project 3.

Signed: Daniel Ying dty16, Kyle VanWageninge kjv48

Workload:

Daniel Ying: Formatted the report, wrote the report in latex code. Compiled the results and answers onto the report.

Kyle VanWageninge: Worked out how to solve question 1, coded basic agents and improved agents. Gathered results for each agent.

Together: Discussed how we would like to implement our improved agent. Discussed the best way to take results for the agents. Talked about answers to the problems and observations of our results.

Problem 1

1) Given observations up to time_t(Observations_t), and a failure searching Cell_j (Observations_t+1= Observations_tFailure 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 and Failure in Cell_j)

ANSWER:

- A = Target in Cell_i,
- B = Failure in Cell_j,
- obs = observation_t

$$\frac{P\quad(A \wedge B \mid obs)}{P(B \mid obs)}$$

$$\Rightarrow \frac{P(B \mid A, obs)P(A \mid obs)}{P(B \mid obs)}$$

$\Rightarrow P(B \mid obs) = P(B \wedge A \mid obs) + P(B \wedge \neg A \mid obs)$, using marginalization

$\Rightarrow P(B \mid A, obs)P(A \mid obs) + P(B \mid \neg A, obs)P(\neg A \mid obs)$, conditional factoring

$\Rightarrow P(B \mid A)P(A) + P(B \mid \neg A)P(\neg A)$, using simplification and observation model

$$\Rightarrow \frac{P(B \mid A)P(A)}{P(B \mid A)P(A) + P(B \mid \neg A)P(\neg A)}$$

$$\Rightarrow \frac{P(A)}{P(A) + P(B \mid \neg A)(\neg A)},\ \text{the } P(B \mid \neg A) \text{ was gotten by the calculation below:}$$

$$P(B \mid \neg A) = \frac{P(B \wedge \neg A)}{P(\neg A)} = \frac{P(B) - P(A)P(B \mid A)}{P(\neg A)},\ \text{in which:}$$

$P(B) = P(\text{in cell\_j})P(B \mid \text{in cell\_j}) + P(\text{not in cell\_j})P(B \mid \text{not in cell\_j})$

Final Answer: $\dfrac{P(A)}{P(A) + P(B \mid \neg A)P(\neg A)}$

Exception:
If Conditions: B = (failure in cell_i) ,
A = (in cell_i) then $P(B \mid \neg A) = 1$
because: if the target is ¬ (in cell_i), failure will be 100%
thus the final equation would be:

$$\Rightarrow \frac{P(A)P(B \mid A)}{P(A)P(B \mid A) + (1)P(\neg A)}\ \text{, where } P(B \mid A) \text{ is a false negative of cell type}$$

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:

- P(Found in Cell_i | Observations_t)

  ANSWER:

  - A = Found in Cell_i,
  - B = In Cell_i
  - obs = observation_t

  $\Longrightarrow P(A \wedge B \mid obs) + P(A \wedge \neg B \mid obs)$, using marginalization,

  $\Longrightarrow P(A \mid B)P(B)+P(A \mid \neg B)P(\neg B), P(A \mid \neg B) = 0$ by no false positive; using conditional factoring

  $\Longrightarrow P(A \mid B)P(B)+)$

  $\Longrightarrow$ (1 – false negative for cell type)$P(B)$

  Final Answer: (1- false negative for cell type) P(B)

Problem 3

Consider the following situation. Your agent is dropped into the map at a random location and wants to find the target as quickly as possible. At every time step, the agent can either a) search the current cell the agent is in, or b) move to one of the immediate neighbors (up/down/left/right). We can consider the following basic agents:

- Basic Agent 1: Iteratively travel to the cell with the highest probability of containing the target, search that cell. Repeat until target is found.
- Basic Agent 2: Iteratively travel to the cell with the highest probability of finding the target within that cell, search that cell. Repeat until the target is found.

For both agents, ties in probability between cells should be broken based on shortest distance (minimal Manhattan distance), and broken at random between cells with equal probability and equal shortest distance. The final performance of an agent is taken to be 'total distance traveled' + 'number of searches', and we want this number to be as small as possible.

Generate 10 maps, and play through each map (with random target location and initial agent location each time) 10 times with each agent. Which agent is better, on average?

ANSWER:

Agent 1:

We started agent 1 with the base probabilities in each cell (1/area). We then search the cell where the agent was placed at, because currently all probabilities are the same and the cell where the agent is at has a Manhattan distance of 0.

Under the condition where A = target in cell_i and B = failure in cell_i (as seen in the Conditions line in problem 1), we would use the equation from question 1 to update the current cell after a failure (false negative rate) appears. Then, since the probability of the searched cell decreased, the probability of the other cells would go up, though not uniformly. Here is when we use the original final equation from problem 1, where A = (target in cell_i) and B = (failure in cell_j), to update the belief of each cell based on the list of observations collected. After updating the belief, we then find the cell with the highest probability of the target being in the cell. If there is a tie, the search will be broken up based on shortest distance i.e the minimal Manhattan distance, and broken at random between cells with equal probability and equal shortest distance. After a cell is chosen, we will search it and repeat the implementation.

Finally after the agent has found the target, the agent will return the total score which is the 'total distance traveled' + 'number of searches'.

Agent 2:

Like agent 1, we started agent 2 with the base probabilities. But this time,before searching a cell, we add a step to find the cell with the highest probability of finding the target in that cell (based on current observations in the cell).

Here is where the equation in problem 2's final answer comes into play. We will use this on each of the cells to figure out what is the likelihood of finding that target in that cell. Once these probabilities are calculated they will be put into a second array, which will then be search for the highest probability of finding the target. Then, ties in probability will be broken up

based on the shortest distance (like agent 1), and broken at random between cells of equal probability and distance.

Then we will choose the cell the agent will move to and search. When the cell gets searched, it will update the beliefs the same way as agent 1 did. After the belief update, we will go through our implementation of finding the target in cell_i as previously mentioned.

Finally, after the agent has found the target, the agent will return the total score which is the 'total distance traveled' + 'number of searches'.

Results:

We ran each map 50 times, each with a different starting agent and target location. Both agents (1 and 2) used the same starting points in each run.

Map1: Score Comparison of Agent 1 and 2 on 4 target lands types.				
Land Type	Flat	Hill	Forest	Cave
Agent 1	19482	23313	46477	78689
Agent 2	3806	15680	16883	62658

Map2: Score Comparison of Agent 1 and 2 on 4 target lands types.				
Land Type	Flat	Hill	Forest	Cave
Agent 1	29112	37483	95495	65227
Agent 2	8631	3152	36143	85779

Map3: Score Comparison of Agent 1 and 2 on 4 target lands types.				
Land Type	Flat	Hill	Forest	Cave
Agent 1	2278	34566	56586	45295
Agent 2	843	24277	35857	105992

Map4: Score Comparison of Agent 1 and 2 on 4 target lands types.				
Land Type	Flat	Hill	Forest	Cave
Agent 1	13402	27728	68650	81730
Agent 2	6303	3940	18412	114665

Map5: Score Comparison of Agent 1 and 2 on 4 target lands types.				
Land Type	Flat	Hill	Forest	Cave
Agent 1	2954	30772	48705	115425
Agent 2	5708	14256	24922	57745

Map6: Score Comparison of Agent 1 and 2 on 4 target lands types.				
Land Type	Flat	Hill	Forest	Cave
Agent 1	532	82470	25149	101871
Agent 2	993	4667	23527	96526

Map7: Score Comparison of Agent 1 and 2 on 4 target lands types.				
Land Type	Flat	Hill	Forest	Cave
Agent 1	2311	24845	37688	97619
Agent 2	831	4459	28429	94215

Map8: Score Comparison of Agent 1 and 2 on 4 target lands types.				
Land Type	Flat	Hill	Forest	Cave
Agent 1	35400	38471	37800	85318
Agent 2	921	7076	13927	80145

Map9: Score Comparison of Agent 1 and 2 on 4 target lands types.				
Land Type	Flat	Hill	Forest	Cave
Agent 1	31528	10918	33586	70866
Agent 2	7647	4759	15106	72289

Map10: Score Comparison of Agent 1 and 2 on 4 target lands types.				
Land Type	Flat	Hill	Forest	Cave
Agent 1	34576	57660	68107	95550
Agent 2	976	7290	35860	82373

These results show that, with agent 2, using the probability to find the target in a given cell should produce better search results, as compared to agent 1's searching a cell it thinks may have the target would most likely lead to a bad search. But there are exceptions. We noticed that there are cases in which agent 2 produced results worse that agent 1. We believe this is because there is no guarantee that results would be consistent for the calculations are based on false negative rates. The target could be in any landscape and the agent could continue to fail to find it in the search. This would also explain the relatively large deviations in scores between the two agents, which is also why we ran 50 different agent and target locations on a single map and recorded the average of each scores based on the terrain type. In the end this allowed us to see the average performance of the agent in each landscape type on each specific map, which, expectedly, most of the times agent 2 performed better than agent 1.

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?

ANSWER:

PART I:

ACRONYM: S.I.T.H (Search Intensely, Try Harder)

After reviewing the scores of agent 1 and 2, we observed that searching a cell only a single time in each step was still returning a relatively high score. This is most likely resulted by situations in which the agent searched the correct cell for the target, yet returned a failure, thus taking a little more time to return back to that target cell. Thus in our improved agent, we will search each cell differently based on it landscape type.

For example, a flat landscape cell should only get searched once, because of the 0.1 false negative, hill has 0.3 false negative rate, while Forest is 0.7 and Cave is 0.9. Thus the flat landscape is searched 1 time, the Hill is searched 3 times, the Forest is searched 7 times, and finally the Cave is search 9 times. The difference in times of searches in the landscape greatly improved the searching success across all terrain types as shown in the Results. The difference in the search times allowed the probability of the cell containing the target to decrease in failing to find the target.

The improved agent is useful because depending on the landscape we can avoid unnecessary travel. Say the target was placed in a cave, agent 1 or 2 will travel to that cave, search it, and most likely to fail due to the Cave's high false negative rate. After which the agent will have to return to that cell again and again would continue to fail, thus making the two agents relatively inefficient compared to the improved agent.

So, with the improved agent, we are searching the cell differently based on landscape, thus decreasing the amount of times the agent would get to a cell and leave the cell before enough searches are conducted. So the improved agent would save much more traveled distance compared to agents 1 and 2.

RESULTS:

We ran each map 50 times, each with a different starting agent and target location. Both agents (1 and 2) used the same starting points in each run.

Map1: Score Comparison of Agent 1, 2, and Improved Agent on 4 target lands types.				
Land Type	Flat	Hill	Forest	Cave
Agent 1	2435	15289	57263	74170
Agent 2	6844	3234	26705	77307
Improv. Agent	2182	3728	12560	20876

Map2: Score Comparison of Agent 1, 2, and Improved Agent on 4 target lands types.				
Land Type	Flat	Hill	Forest	Cave
Agent 1	14913	36472	58516	72887
Agent 2	5166	7406	29534	95782
Improv. Agent	4101	3736	9745	29477

Map3: Score Comparison of Agent 1, 2, and Improved Agent on 4 target lands types.				
Land Type	Flat	Hill	Forest	Cave
Agent 1	2627	28116	63093	68225
Agent 2	1384	14538	16954	79743
Improv. Agent	3298	3985	14767	20075

Map4: Score Comparison of Agent 1, 2, and Improved Agent on 4 target lands types.				
Land Type	Flat	Hill	Forest	Cave
Agent 1	41990	79363	22779	58249
Agent 2	944	10045	20832	67741
Improv. Agent	3663	3199	7649	20516

Map5: Score Comparison of Agent 1, 2, and Improved Agent on 4 target lands types.				
Land Type	Flat	Hill	Forest	Cave
Agent 1	51897	48405	77485	46560
Agent 2	2848	7817	34208	65432
Improv. Agent	1997	10502	11517	22059

Map6: Score Comparison of Agent 1, 2, and Improved Agent on 4 target lands types.				
Land Type	Flat	Hill	Forest	Cave
Agent 1	3245	41656	80113	67177
Agent 2	18096	9109	9306	66111
Improv. Agent	1082	6303	7668	18830

Map7: Score Comparison of Agent 1, 2, and Improved Agent on 4 target lands types.				
Land Type	Flat	Hill	Forest	Cave
Agent 1	2416	46842	86881	66202
Agent 2	806	11940	66472	50935
Improv. Agent	2195	8099	8093	20650

Map8: Score Comparison of Agent 1, 2, and Improved Agent on 4 target lands types.				
Land Type	Flat	Hill	Forest	Cave
Agent 1	11535	48882	39606	86401
Agent 2	14402	12734	41061	45649
Improv. Agent	846	6932	7039	24578

Map9: Score Comparison of Agent 1, 2, and Improved Agent on 4 target lands types.				
Land Type	Flat	Hill	Forest	Cave
Agent 1	14970	30376	72112	53683
Agent 2	965	8636	19757	93402
Improv. Agent	5381	3582	9551	23178

Map10: Score Comparison of Agent 1, 2, and Improved Agent on 4 target lands types.				
Land Type	Flat	Hill	Forest	Cave
Agent 1	2109	27869	39337	47370
Agent 2	3735	13598	10690	85736
Improv. Agent	4031	3476	15424	16477

Looking at the results of our improved agents, it shows that improved agent produced better results than agent 1 and 2, because improved agents uses a better probability search and does a more thorough search for each cell based on its landscape type. This proves that a good deal of extra distance traveled by agent 1 and 2 can be avoided based on the multiple searches (which depended on the landscape type being searched), allowing the target to be found quicker and more efficiently. For each map, the average of the improved agent was better than the other two agents most of the times (more times than agent 2 outperforming agent 1).

PART II:

Under such conditions, we can make our agent even better by changing how the agent decide the number of searches to do using the MDP model. This would show us, after a search action and a potential failure, what the state would look like after another portential search.

For example, we search a cave cell. We will search it once and obtain our result, then we would use the MDP model to look one step ahead. This would be accomplished by finding which step would find another cell that: 1) have the highest probability to finding our currently cell and 2)its distance is the closest to our current cell. This could either search the same cell again, since it has the highest probability and lowest distance, or it could leave the case after conduction a single search and more to the newest lowest cost cell.

This MDP could be used to evaluate multiple searches as well, instead on only one search ahead. It could potentially look two searches ahead and find the best action to take with the lowest cost and the highest probability of finding the target.