CS 520: Assignment 3 - Probabilistic Search (and Destroy)

Haoyang Zhang, Han Wu, Shengjie Li, Zhichao Xu

November 20, 2018

1 Introduction, group members and division of workload

In this group project, we implemented a minesweeper solver that far exceeded our expectation. Not only can our program solve the normal minesweeper, but also it can solve minesweeper with inaccurate information. Our program also has a gorgeous GUI and can visualize the progress of solving minesweeper by animation.

Name RUID	Workload
Haoyang Zhang 188008687	Implemented the minesweeper solver. Finished the writing of report for most of the questions. Wrote *Solution Algorithm Explanation.html* and *Uncertainty Explanation.html* which are two documents about our algorithm
Han Wu 189008460	Ran tests and generated figures for question 2.4. Finished the writing of report for question 2.4
Shengjie Li 188008047	Designed and implemented the GUI of our program. Implemented a function that can generate animation of the progress of solving minesweeper. Finished the format design of whole report using LaTeX.
Zhichao Xu 188008912	Proofread the report. Ran tests and generated figures for question 2.5 and question 4.1. Finished the writing of report for question 2.5 and question 4.1.

2 A Stationary Target

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

$$\mathbb{P}(\text{Target in Cell}_i|\text{Observations}_t \wedge \text{Failure in Cell}_j). \tag{1}$$

When $i \neq j$,

$$\begin{aligned}&\mathbb{P}(\text{Target in Cell}_i|\text{Observations}_t \wedge \text{Failure in Cell}_j)\\ &= \alpha\mathbb{P}(\text{Target in Cell}_i|\text{Observations}_t)\\ &= \alpha \cdot a_{it}.\end{aligned}$$

When $i = j$,

$$\begin{aligned}
&\mathbb{P}(\text{Target in Cell}_j|\text{Observations}_t \wedge \text{Failure in Cell}_j) \\
&= \alpha\mathbb{P}(\text{Target in Cell}_j|\text{Observations}_t) \cdot \mathbb{P}(\text{Target not found in Cell}_j|\text{Target in Cell}_j) \\
&= \alpha \cdot a_{jt} \cdot q. \\
\alpha &= \frac{1}{\sum_{i \neq j}(\alpha \cdot a_{it}) + \alpha \cdot a_{jt} \cdot q} = \frac{1}{1 - a_{jt}(1-q)}. \\
a_{i0} &= \frac{1}{2500} \text{ for } i = 0, \ldots, 2499
\end{aligned}$$

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:

$$\mathbb{P}(\text{Target found in Cell}_i|\text{Observations}_t)? \tag{2}$$

$$\begin{aligned}
&\mathbb{P}(\text{Target found in Cell}_i|\text{Observations}_t) \\
&= \mathbb{P}(\text{Target in Cell}_i|\text{Observations}_t)(1 - \mathbb{P}(\text{Target not found in Cell}_i|\text{Target in Cell}_i)) \\
&= a_{it} \cdot (1-q) \\
a_{i0} &= \frac{1}{2500} \text{ for } i = 0, \ldots, 2499
\end{aligned}$$

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?

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.

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.

3 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** $\times$ **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.