# project3: Probabilistic Search

Bingyu Xin

November 2020

## 1 Update the belief state

We define Events:

$$T_i = \{Target\ in\ Cell_i\}$$

$$O_t = \{Observations_t\}$$

$$F_j = \{Failure\ in\ Cell_j\}$$

Given that ,

$$Belief_i = P(T_i|O_t)$$

$$P(F_i|T_i) = FNR$$

we have ,

$$\begin{aligned} P(T_i|O_tF_j) &= \frac{P(O_tF_j|T_i)P(T_i)}{P(O_tF_j)} \\ &= \frac{P(O_t|T_i)P(F_j|O_tT_i)P(T_i)}{P(O_t)P(F_j|O_t)} \\ &= Belief_i \frac{P(F_j|O_tT_i)}{P(F_j|O_t)} \\ &= \begin{cases} Belief_i \cdot FNR & i = j \\ Belief_i & i \neq j \end{cases} \end{aligned}$$

(1)

## 2 Update Confidence state

we have,

$$\begin{aligned} P((1-F_i)|O_t) &= P((1-F_i)|O_tT_i)P(T_i|Q_t) \\ &= (1-FNR) \cdot Belief_i \end{aligned}$$

(2)

## 3 Belief vs Confidence

For Rule 1, it finds the max value of the belief array to check; for Rule2, it finds the max value of the Confidence array to check;

In our experiments, we set map size as 10 and 20 separately to calculate the average searches, and we find that Rule2 performs better than Rule1 as shown in 1, because cells with high confidence must be cells with high belief, but not vise versa.

Table 1: Average searches for different Rules

| dim | 10 | 20 |
|---|---|---|
| Rule 1 | 499 | 1702 |
| Rule 2 | 126 | 626 |

## 4 Modified problems

We implement three basic Agent in our code, the results are shown as 2. Agent 1 and Agent 2 are similar to last experiment, but now we prefer to choose point closer to our current position to minimize the motion actions; Agents 3 has the best performance because it uses heuristic utility to make compromise between manhattan distance and confidence value. We have tried some exponential cost function for improved agent but it doesn't beat agent 3.

Table 2: Average searches for different Rules

| dim | 10 | 20 |
|---|---|---|
| Agent 1 | 3803 | 27417 |
| Agent 2 | 698 | 4732 |
| Agent 3 | 335 | 1039 |