

# Artificial Intelligence: HMM

In this project, we aim to implement Localization with the HMM algorithm.

## 1 Instructions

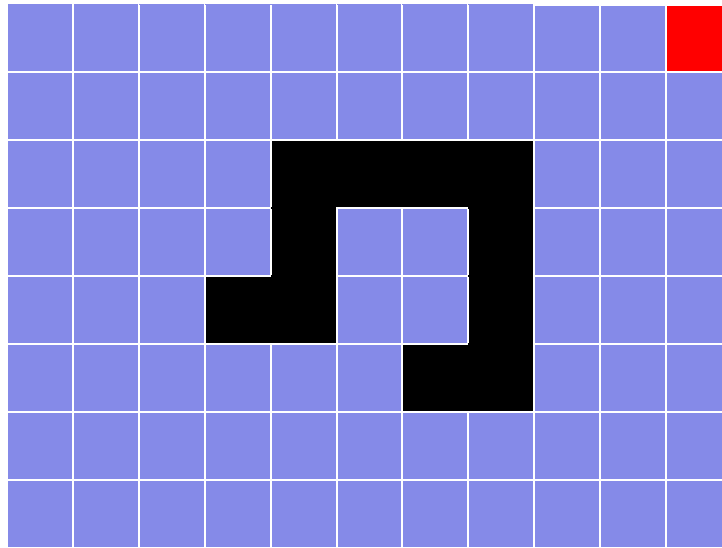


Figure 1: Windy Maze

We assume that a robot aims to locate itself in the windy maze defined in P1. The maze map is shown in Figure 1 for your convenience. The robot will perform two kinds of actions: sensing and moving.

**Sensing** In a square, the robot will sense the four directions to see if there is an obstacle in this direction. We assume that the whole maze is surrounded by obstacles and the black squares are also obstacle. However, the sensing is not perfect. We assume that the robot can detect the obstacle with 90% if there is and might mistake an open square as a obstacle with 5%. The detections in all directions are done independently.

**Moving** In the windy situation, the robot can drift to the left or the right with probability 0.1. If the drifting direction is an obstacle, it will be bounced back to the original position. For example, in the square of left bottom, if the robot moves northward, it will reach the square to the north with 80% and reach the square to the east with 10% and be bounced back to the original position with 10%.

We assume that the robot initially stays in one open square, but it doesn't know its exact location except that it knows that it can't be in any obstacle square. Then the robot performs the following sequence of actions:

- 1.Sensing: (-,-,-,-)
- 2.Moving northward
- 3.Sensing: (-,-,-,-)
- 4.Moving northward
- 5.Sensing: (-,-,0,-)
- 6.Moving northward
- 7.Sensing: (-,-,-,-)
- 8.Moving eastward
- 9.Sensing: (-,-,0,0)
- 10.Moving northward
- 11.Sensing: (-,-,0,-)

where (W,N,E,S) indicates the observation at Directions (Westward, Northward, Eastward, Southward), respectively. "-" indicates no obstacle is observed and "o" indicates an obstacle is observed.

You are expected to report the posterior probability of the current robot location at each square after each action. The prior probability at each square is shown as follows (1.30 means 1.30%):

1.30	1.30	1.30	1.30	1.30	1.30	1.30	1.30	1.30	1.30	1.30
1.30	1.30	1.30	1.30	1.30	1.30	1.30	1.30	1.30	1.30	1.30
1.30	1.30	1.30	1.30	0.00	0.00	0.00	0.00	1.30	1.30	1.30
1.30	1.30	1.30	1.30	0.00	1.30	1.30	0.00	1.30	1.30	1.30
1.30	1.30	1.30	0.00	0.00	1.30	1.30	0.00	1.30	1.30	1.30
1.30	1.30	1.30	1.30	1.30	1.30	0.00	0.00	1.30	1.30	1.30
1.30	1.30	1.30	1.30	1.30	1.30	1.30	1.30	1.30	1.30	1.30
1.30	1.30	1.30	1.30	1.30	1.30	1.30	1.30	1.30	1.30	1.30

Please use the same format to show all posterior probabilities. If the probability is 23.46%, use 4-character format 23.5 to align up all data.

## 2.

In the above, we focus on filtering and prediction in HMM. In the following, we aim to address smoothing in HMM. After all the 11 actions including 5 moves, the robot wants to perform smoothing to improve the estimation of all previous 5 locations. Please implement the smoothing function and show the improved estimation of location probabilities for all previous 5 locations. Please refer to Subsection 15.2.2 in the textbook (on Page 574 of Ed. 3) for the smoothing algorithm. Compare the results before smoothing with the results after smoothing to see how much uncertainty has been reduced.

We use Information Entropy<sup>1</sup> to measure the uncertainty of the location. For  $n$  possible locations, we have  $n$  probabilities  $p_i, i = 1, 2, \dots, n$ . Then the location entropy is  $-\sum_{i=1}^n p_i \log_2 p_i$ . The smaller the entropy is, the lower uncertainty the location is.