Robot Localization with HMM

Tsvetomila Mihaylova

1 Introduction

The project implements robot localization with a hidden Markov model. We are given a square grid with a specified size (20x20). The grid contains obstacles which the robot cannot move into. The robot has sensors which can detect whether there is an obstacle in the 4 neighbouring cells: North, East, South, West. The information from the robot sensors represent the observed states of the system. The robot location in the grid represents the hidden states. Therefore, the system has $20x20 = 400$ **hidden states** and $4^2 = 16$ **observed states** (all the possible variants of presence or absence of an obstacle in the 4 directions). We treat the grid borders as obstacles.

The robot starts on a random position in the grid and starts moving randomly. For each move, we record the observed state, which is the information from the robot sensors. Based on the observed sequence, at each time step we calculate the probability of the robot being in each cell in the grid. The expectations are that the robot will be visible in the states with high probability. The robot movement is random with equal probability of each possible move. One possible move is defined by the robot position in the grid. An allowed move includes all possible neighbouring cells in direction north, east, south or west of the current robot position, which are not occupied by an obstacle and are not outside of the grid. Stain at the current position is also a possible move.

For calculating the probability of the robot location in the grid, given the sequence of observed states, we use the forward algorithm.

The robot starts moving from a random free cell in the grid and we record its observations for estimating its position. For the initial state, we assume uniform distribution of the possible hidden states. I.e. we multiply the probability for an available cell $1/(gridsize - obstacles)$ by the emission probabilities of the first observation.

Hidden States	$20x20 = 400$	Each cell in the grid is a hidden state.
Observed States	$4^2 = 16$	The possible observations are all possible combinations of $0s$ and $1s$ in the four directions from each cell.

Table 1: System states.

2 Implementation

The system is implemented in Python. The library *numpy* is used for calculations. The library *tkinter* is used for the graphical part of the project.

In the initialization, we specify the grid with the given obstacles and a random free position as the initial robot position in the grid. For each random movement of the robot, the following steps are performed:

1. Perform a random step among the possible moves.
2. Record the current observation.
3. Show the new robot position in the grid.
4. Calculate the probabilities for the possible position in the current step, based on the current observed state and the sequence of all observed states so far.
5. Clear the old possible positions from the grid.
6. Show the new possible positions of the robot in the grid.
7. Keep the possible states in memory in order to clear them for the next step.

3 Demonstration

We experimented with random assignment of obstacles (Figure 1), as well as with manually set obstacles (Figure 2). At the beginning the possible locations of the robot are spread through the whole grid. As the robot makes more moves and has more observations, the possible locations predicted by the model are more concentrated towards its current position.

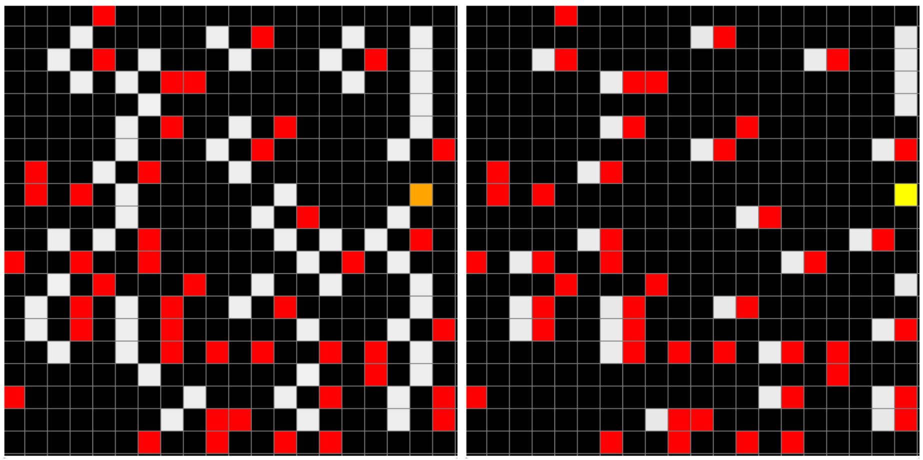

Figure 1: Grid with randomly set obstacles.

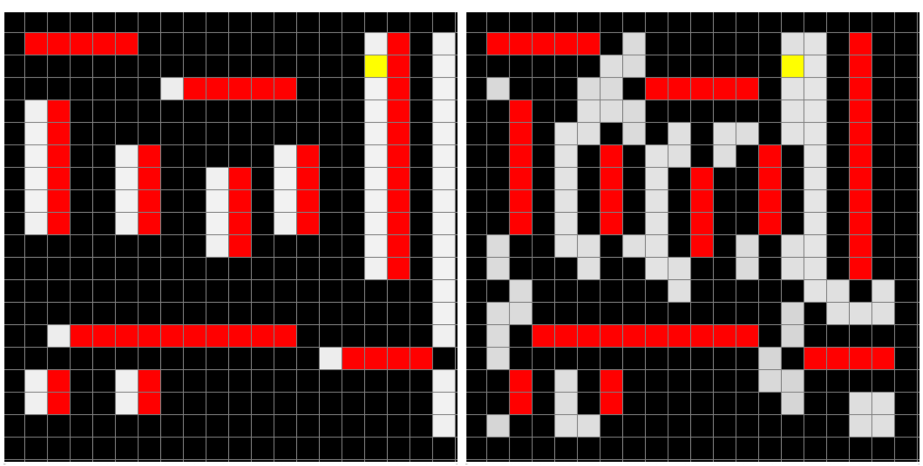

Figure 2: Grid with manually specified obstacles.