Fujian Yan, School of Computing

HMM for Robot Localization

1. Background

In this project, you are assigned to localize the robot in a mini-warehouse (the provided map) with the Hidden Markov Method. Reference: https://dtransposed.github.io/blog/2018/04/01/Robot-Localization/

- **Localization:**

  Robot localization is the process of determining where a mobile robot is located with respect to its environment. Localization is one of the most fundamental competencies required by an autonomous robot as the knowledge of the robot's own location is an essential precursor to making decisions about future actions. In a typical robot localization scenario, a map of the environment is available, and the robot is equipped with sensors that observe the environment as well as monitor its own motion. The localization problem then becomes one of estimating the robot position and orientation within the map using information gathered from these sensors. Robot localization techniques need to be able to deal with noisy observations and generate not only an estimate of the robot location but also a measure of the uncertainty of the location estimate.

- **Hidden Markov Models:**

  The Hidden Markov Model (HMM) is a simple way to model sequential data. The HMM can be used in situations such as the data consists of a sequence of observations, the observations depend (probabilistically) on the internal state of a dynamical system, and the true state of the system is unknown.

2. Environment

Environment Description: The environment Illustration of the agent consists of 16 discrete states. The time is also discrete. At each subsequent time step the robot is programmed to change its position with a probability of 80% and moves randomly to a different neighboring tile. An example of state S4 has shown Illustration. As soon as the robot makes a move, we receive four readings from the sensing system. In the environment Illustration, there are two shelves that are colored in yellow. The shelves will block the agent.

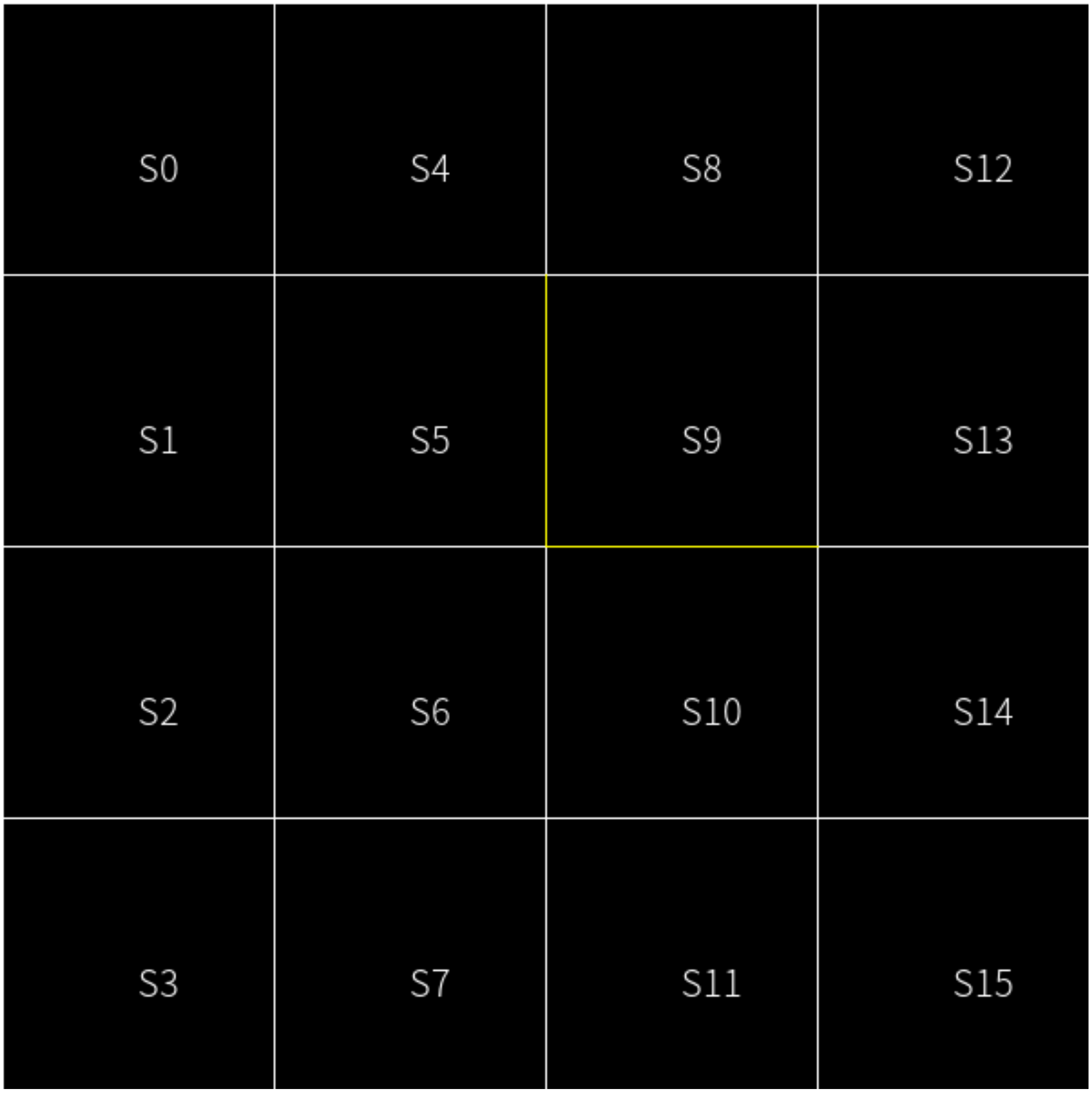


Illustration 1: A sample of map.

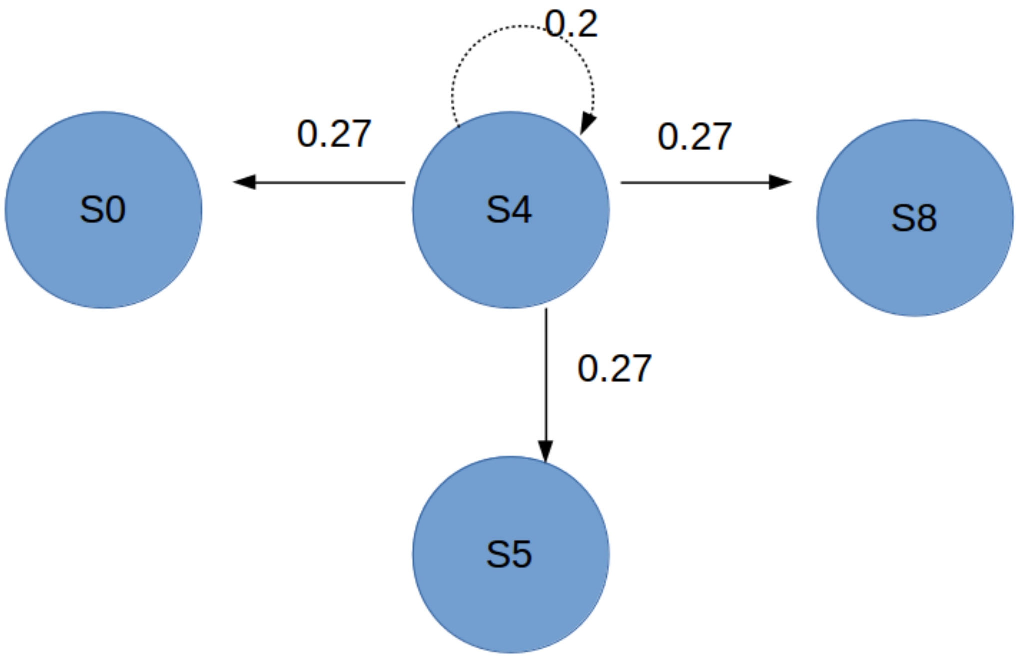


Illustration 2: The probability for robot to move form tile S4.

Sensor: Just as we humans can localize ourselves using senses, robots use sensors. The agent is equipped with the sensing system composed of a compass and a proximity sensor, which detects obstacles in four directions: north, south, east and west. The sensor values are conditionally independent given the position of the robot. Moreover, the device is not perfect, the sensor has an error rate of e = 25. The discrepancy is d, several signals that are different- between the true values for tile i and the actual reading. Assume the robot is at state i, and the probability that the sensor returns reading E=j is based on:

$$O_{i,j} = P(E = j|X = i) = (1 - e)^{4-d} * e^d$$

3. Project Objective and Requirements

Objective:

- Given a random map (4-by-4)
- Given a sequence of sensor readings (5): NW, NW, N, NE, SWE.
- Q1: Find the probability plots for each time step
- Q2: Predict for next two states with given the first three states. (NW, NW, N)

- Q3: What happened when the error is 0.025? Repeat Q1, and Q2 with e = 0.025
- Q4: What happened when the error is 0.5? Repeat Q1, and Q2 with e =0.5.
- Q5: Compare results with Q3 and Q4
- Need to turn in a report.

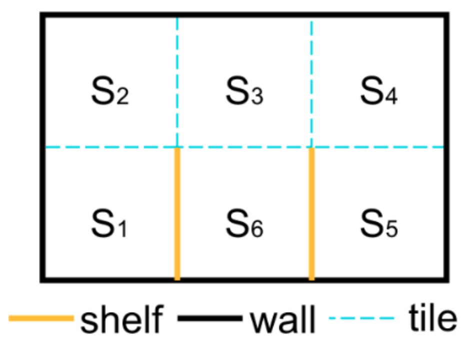
Requirements:

A. Report (50%)
   1. Probability plot for Q1, with the max probability for each timesteps. (10%)
   2. Predicted results for next two steps, and their probability (10%)
   3. Plots, max probability, and predicted results and corresponding probability for new error (10%)
   4. Plots, max probability, and predicted results and corresponding probability for new error (10%)
   5. A paragraph to discuss the observation (10%)

B. Code (50%)

Example:

- The sensor reading: SWE.
- The environment:



- With the error rate e = 0.25, and discrepancies for the whole environment is [0,3,4,3,0,0]
- The observation matrix with the given sensor reading for the whole environment is:

```
[[0.31640625 0.         0.         0.         0.         0.        ]
 [0.         0.01171875 0.         0.         0.         0.        ]
 [0.         0.         0.00390625 0.         0.         0.        ]
 [0.         0.         0.         0.01171875 0.         0.        ]
 [0.         0.         0.         0.         0.31640625 0.        ]
 [0.         0.         0.         0.         0.         0.31640625]]
```

6. Construct the HMM based on the above information and the following equations: (worth 2%)

- filtering- estimation of the state in time X_t, knowing the state X_{t-1} and evidence E_t.

$$f_{1:t} = \alpha * O_t * T * f_{1:t-1}$$

- prediction- filtering without evidence. We make a guess about the $X_t$knowing only state X_{t-1}.

$$f_{1:t} = \alpha * T * f_{1:t-1}$$

where:

- $f_{1:t}$ current probability vector in time *t*
- $f_{1:t-1}$ previous probability vector in time *t-1*
- α normalization constant
- O_t observation matrix for the evidence in time *t*
- T transition matrix

4. Deliverables, Evaluation Criteria and Timeliness

Following are the expected deliverables and the weightage:

#	Deliverable	Grade Weightage	Due Date
1	Functional codes	50%	End-of-Day 03/24/2025
2	Project Report	50%	End-of-Day 03/24/2025

5. Academic Integrity

Students at Wichita State University are expected to uphold high academic standards. WSU will not tolerate a lack of academic integrity. Students are responsible for knowing and following the Student Code of Conduct http://webs.wichita.edu/inaudit/ch8_05.htm and the Student Academic Honesty policy http://webs.wichita.edu/inaudit/ch2_17.htm. When the faculty member determines sanctions are warranted for violations of academic integrity, regardless of severity, the faculty member must report the infraction to the Office of Student Conduct and Community Standards. If you need more information about the process or wish to appeal a decision, please visit! https://www.wichita.edu/about/student_conduct/AcademicDishonesty.php.

6. Development Environment

OS: Linux, OS, and Windows

Development tool (may need): NumPy, Matplotlib