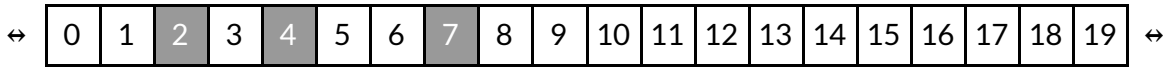
49274 Advanced Robotics (Spring 2019)

Assignment 1 Part 1: Bayes Filter

- Due date: 12/08/2019 (week 4) at 6 pm
- Total marks: 5 (5% of the final mark for the subject)

Introduction

In this assignment you will solve the localisation problem using a discrete Bayes filter. Imagine a robot that lives in a 1-dimensional world shown in the figure below. The world is divided into 20 cells, numbered 0 to 19. The world is circular: the first and last cells are adjacent to each other so that cell 0 is one step ahead of cell 19. There are also three doors in the world, located at cells 2, 4, and 7.



At each time step the robot is located in only one of the cells. We use x_t to denote the position of the robot at time t. E.g. $x_2 = 4$ means that the robot is in cell 4 at time step 2.

Motion probability

The robot can be moved forward by sending a motion command. We use u_t to denote the motion command at time step t. The motion command is either 0 (stay) or 1 (move). E.g:

- Stay: $u_t = 0$
- Move: $u_t = 1$

Ideally the move command would result in the robot moving one cell forward, but in reality there is the possibility that the robot will move two cells forward.

Given a command to move, the probability of the robot moving **one cell** forward is 0.7. Mathematically we write this as:

$$P(x_{t+1} = k + 1 | x_t = k,\ u_t = 1) = 0.7$$

Where k is a cell.

Given a command to move, the probability of the robot moving **two cells** forward is 0.3. Mathematically we write this as:

$$P(x_{t+1} = k + 2 | x_t = k,\ u_t = 1) = 0.3$$

Given a command to move, the probability of the robot moving to **any other cell** is 0. This includes staying in the same cell.

Given **no command to move**, the probability of the robot staying in the **same cell** is 1.0. Mathematically we write this as:

$$P(x_{t+1} = k | x_t = k,\ u_t = 0) = 1.0$$

Given no command to move, the probability of the robot moving to **any other cell** is 0.

Observation probability

There are three doors located at cell 2, 4, and 7. The robot is equipped with a sensor that can detect these doors, which we can use for localisation. At each time step the robot can detect a door ($z_t = 1$) or detect no door ($z_t = 0$).

The door sensor is also imperfect, and there is a possibility of detecting no door when there is a door, or detection a door when there is no door.

Given that there is a door, the probability of detecting a door is 0.8. Mathematically we write this as:

$$P(z_t = 1 | x_t = \{2, 4, 7\}) \ = \ 0.8$$

Given that there is a door, the probability of detecting **no door** is 0.2. Mathematically we write this as:

$$P(z_t = 0 | x_t = \{2, 4, 7\}) \ = \ 0.2$$

Given that there is **no door**, the probability of detecting a door is 0.1. Mathematically we write this as:

$$P(z_t = 1 | x_t \neq \{2, 4, 7\}) \ = \ 0.1$$

Given that there is **no door**, the probability of detecting **no door** is 0.9. Mathematically we write this as:

$$P(z_t = 0 | x_t \neq \{2, 4, 7\}) \ = \ 0.9$$

Bayes Filter

We use a discrete Bayes filter to calculate a probability distribution that reflects of belief about the robot's location. We call this the belief state. The first step in the Bayes filter is initialisation, where we calculate initial values for the belief state. Then at each time step, given a motion command and sensor observation, we perform a motion update, sensing update, and normalisation.

Initialisation

With no information about the robot's location in the world, the probability of the robot being located in a cell is the same for every cell. So:

$$P(x_t = k) = \frac{1}{number\ of\ cells}$$

For example, if we have 5 cells our belief state will be:

Position (k)	0	1	2	3	4
$P(x_t = k)$	0.2	0.2	0.2	0.2	0.2

Motion update

After the motion update, the probability of each cell in the belief state is the sum of the probability of every cell multiplied by the probability of moving from the previous cell to the current cell, given a motion command input. Mathematically we write this as:

$$\hat{P}(x_{t+1} = k|u_t) = \sum_{i=0}^{19} P(x_t = i) \times P(x_{t+1} = k|x_t = i,\ u_t)$$

Where k is the current cell, and i is the previous cell. This is actually a pseudo probability (indicated with a hat) because the belief state is not yet normalised.

For example, assuming that the robot was asked to move ($u_t = 1$) we can determine the probability that the robot is in cell 2 at time step $t + 1$. We need to sum all the possible ways that the robot could take to move to cell 2. Given that the robot was asked to move, there are only two ways that the robot could be in cell 2:

- The robot started in cell 0 and moved 2 steps forward
- The robot started in cell 1 and moved 1 step forward

So the probability of being in position 2 at time step $t + 1$ is:

$$\hat{P}(x_{t+1} = 2|u_t = 1) = P(x_t = 0) \times P(x_{t+1} = k + 2|x_t = k, u_t = 1) + P(x_t = 1) \times P(x_{t+1} = k + 1|x_t = k, u_t = 1)$$

Using the belief state from the initialisation example, and the motion probabilities stated previously, the equation becomes:

$$\hat{P}(x_{t+1} = 2|u_t = 1) = (0.2 \times 0.3) + (0.2 \times 0.7)$$

The motion update is performed on every cell.

Observation update

The observation update applies observation probability to the cell probabilities calculated by the motion update. The probability of a cell after the observation update is the probability of the cell after the motion update multiplied by the probability of being in the cell given the observation input. Mathematically we write this as:

$$\hat{P}(x_{t+1} = k|u_t, z_t) = \hat{P}(x_{t+1} = k|u_t) \times P(z_t|x_{t+1} = k)$$

The observation update is performed on every cell.

Normalisation

Once you have applied the motion and observation updates you need to normalise the pseudo probabilities. After normalisation the sum of the probabilities in the belief state is 1.

To normalise, calculate the sum of all (pseudo) probabilities, then divide each by the sum.

Sum of all probabilities:

$$\eta = \Sigma\, \bar{P}(x_{t+1} = k|u_{t,} z_{t+1})$$

Normalised probability for each cell:

$$P(x_{t+1} = k|u_t,\ z_{t+1}) = \frac{\bar{P}(x_{t+1}=k|u_t, z_{t+1})}{\eta}$$

Your task

Your task is to complete a number of functions in the template code provided on UTS Online. There are 5 functions that need to be completed, each worth 1 mark:

- *motionProbability*
- *observationProbability*
- *normaliseState*
- *initialiseState*
- *updateState*

The program starts with the main function on line 95 of the template code. You do not need to modify anything in the main function, but it would be a good idea to understand what the function is doing, so you know how the other functions are used.

The main function:

- Tests the *motionProbability* function with several different values
- Tests the *observationProbability* function with several different values
- Tests the *normaliseState* function with a vector of 5 values

- Creates a vector of 20 values named *state*, calls the *initialiseState* function, then calls the *updateState* function multiple times with various arguments.

The state vector is printed before and after initialisation, and after each use of *updateState*. We will not give you the expected values, you should look at the values and decide if they make sense.

Compiling and running the program

If you are using the UTS computer labs you first need to run:

```
singularity shell /images/singularity_containers/ros-melodic-ar.sif
```

when you open a terminal.

Extract the template code into a folder and open the folder in a terminal. To build run the commands:

```
mkdir build
cd build/
cmake ../
make
```

After you have made changes to your code you can build again just by executing "make".

To run the program run the command "./bayes_filter" in the terminal.