49274: Advanced Robotics	**Spring 2018**

Assignment 1 - Part 1: Bayes Filter

Total Marks: 2 marks

University of Technology Sydney (CAS) Due date: **week 4 (16 Aug. 2018)**

1 Introduction

In this assignment, you will solve the localization problem using the *discrete* Bayes filter.

1.1 Scenario

Imagine a robot that lives in a 1-dimensional world as in Fig. 1. The world is divided into 20 cells as a circle where the cell 20 is connected before cell 1. Each cell has a unique label $i \in \{1, \dots, 20\}$. At each time step, our robot is located at (exactly) one of the cells. We use x_t to denote the position of the robot at time t. For example $x_2 = 4$ means that our robot was in the 4^{th} cell at time $t = 2$.

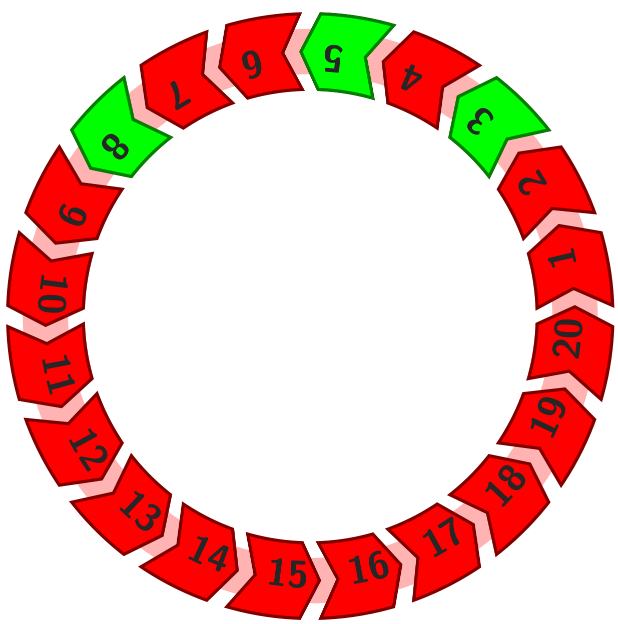

Figure 1: The 1-dimensional world that our robot lives in. There are 3 identical doors located at $\mathbf{3}^{\text{rd}}$, $\mathbf{5}^{\text{th}}$ and $\mathbf{8}^{\text{th}}$ cell. The robot can move forward along the arrows by 1 or 2 cell if control signal $u_t = 1$. The robot will stay still if control signal $u_t = 0$. The doors can be detected by an on-board sensor unreliably-.-, which means the sensor may say there is a door here but actually there isn't a door there, or vice versa.

Motion Uncertainty

The robot can execute a motion command at any time step. We use u_t to denote the motion command given to the robot at time step t. The motion command (a.k.a. control input) is either 0 or 1:

1. $u_t = 0$ means: stay
2. $u_t = 1$ means: move one cell forward

For example, under *perfect* conditions, if $x_3 = 4$ and $u_3 = 1$, then $x_4 = 5$. **But** in practice, due to various reasons such as wheel slippage, our robot may perform differently when $u_t = 1$:

- $x_{t+1} = x_t + 1$ with **probability** 0.7 (correct behaviour)
- $x_{t+1} = x_t + 2$ with **probability** 0.3
- The **probability** of robot going to any other cell is 0

Similarly for $u_t = 0$ we have:

- $x_{t+1} = x_t$ with **probability** 1 (correct behaviour)
- The **probability** of robot going to any other cell is 0

Remark: Mathematically we can write:

- $P(x_{t+1} = k + 1 | x_t = k, u_t = 1) = 0.7$
- $P(x_{t+1} = k + 2 | x_t = k, u_t = 1) = 0.3$
- $P(x_{t+1} = k | x_t = k, u_t = 0) = 1$

Remark: Note that $0.7 + 0.3 + 0 = 1$, and $1 + 0 = 1$.

Observation Uncertainty

There are 3 identical doors in our world. They are located at the **3**$^{\text{rd}}$, **5**$^{\text{th}}$ and **8**$^{\text{th}}$ cell, respectively. Believe it or not, this is our "map" in this example!

The robot is equipped with a sensor that can detect doors. At each time step robot looks at the current cell using its sensor. At time step t sensor returns z_t (our *observation* at time t). $z_t = 1$ means that our sensor has detected a door in the *current* cell, while $z_t = 0$ means no door is detected.

Unsurprisingly our sensor is not perfect either:

1. if there is a door in the current cell x_t (reality):
   - $z_t = 1$ with **probability** 0.8 (correct sensor reading)
   - $z_t = 0$ with **probability** 0.2
2. if there is *not* a door in the current cell x_t (reality):
   - $z_t = 1$ with **probability** 0.1
   - $z_t = 0$ with **probability** 0.9 (correct sensor reading)

Remark: Mathematically we can write:

- $P(z_t = 1 | x_t \in \{3, 5, 8\}) = 0.8$
- $P(z_t = 0 | x_t \in \{3, 5, 8\}) = 0.2$
- $P(z_t = 1 | x_t \notin \{3, 5, 8\}) = 0.1$
- $P(z_t = 0 | x_t \notin \{3, 5, 8\}) = 0.9$

Prior Belief

We might have some information about the initial position of the robot at $t = 0$. This prior knowledge can be represented as a probability distribution. For example we might know that

$$P(x_0 = 1) = 0.8, \quad P(x_0 = i) = (1 - 0.8)/19 = \frac{1}{95} \quad (i \neq 1). \tag{1}$$

which means the robot has a huge probability 0.8 lying at cell 1, while the probability lying at other cells are equal.

If our prior belief is a **uniform probability distribution**, it means that we do not have any *information* about the initial position of the robot. In this case we have:

$$P(x_0 = 1) = P(x_0 = 2) = \cdots = P(x_0 = 19) = P(x_0 = 20) = \frac{1}{20} = 0.05 \tag{2}$$

We use $bel(t)$ to denote our belief at time t. **Belief is the probability distribution of the robot being located at each cell**. In this example, belief is a discrete probability distribution represented by a probability mass function over those 20 cells. Probability distribution in Eq. (1) and probability distribution in Eq. (2) are two examples of belief.

Objective

Given a *prior belief* and a sequence of control inputs (u_t) and observations (z_t) we want to estimate the location of the robot at each time step. In other words, we want to compute the **probability** of robot being located at each cell at any time step.

1.2 Discrete Bayes Filter

At given time t, the location of the robot is described by a discrete probability distribution reflecting our confidence for the robot being at each cell. The probability distribution is described by a probability mass function (PMF) which is denoted by $bel(t)$.

Suppose we have computed our **belief** $bel(t)$ (i.e., probability of the robot being at each cell) at time t. Now given new control input (u_t) and new observation (z_{t+1}), we aim to update our belief and compute $bel(t+1)$ using Bayes filter

$$bel(t) \oplus u_t \oplus z_{t+1} \overset{\text{Bayes filter}}{\longmapsto} bel(t+1)$$

Note that our initial belief $bel(0)$ is a probability distribution of the robot initial position, given/guessed by our prior knowledge. (You have to implement $bel(0)$ in **Q1** and **Q2**).

The discrete Bayes filter consists of two recursive steps:

1. **Prediction**: In this step, we use the new control input u_t and our previous belief $bel(t)$ we compute/predict the position of robot at time $t+1$, denoted by $\overline{bel(t+1)}$ which stands for the belief after only incorporating the control data u_t (note z_{t+1} is not incorporated at this stage):

   $$bel(t) \oplus u_t \longmapsto \overline{bel(t+1)}$$

   Here $\overline{bel(t+1)}$ is our *prediction*, i.e., our belief **before** using the new observation. In order to obtain our prediction, for any cell k we have to compute:

   $$P(x_{t+1} = k|u_t) = \sum_{i=1}^{20} P(x_t = i)P(x_{t+1} = k|x_t = i, u_t). \tag{3}$$

   The probabilities $P(x_t = i)$ over all the cells i at time t constitute $bel(t)$.

   $P(x_{t+1} = k|u_t)$ is the probability that the robot lies in position k at time $t+1$ after considering control u_t. The probabilities $P(x_{t+1} = k|u_t)$ over all the cells k at time $t+1$ constitute $\overline{bel(t+1)}$.

   Intuitively, Eq. (3) means: at time $t+1$, to get the probability of the robot being at position k, we must consider all the possible positions at time t where the robot can move to k by applying a control signal u_t.

2. **Update**: In this step, we fuse our observation z_{t+1} to correct the belief $\overline{bel(t+1)}$ and obtain belief $bel(t+1)$ which stands for the belief after incorporating both the control u_t and observation z_{t+1}.

   $$\overline{bel(t+1)} \oplus z_{t+1} \longmapsto bel(t+1)$$

   We update the probability of the k^{th} (for $k = 1, \ldots, 20$) cell by Bayes Rule

   $$P(x_{t+1} = k|u_t, z_{t+1}) = \frac{1}{\eta} \cdot P(z_{t+1}|x_{t+1} = k)P(x_{t+1} = k|u_t) \tag{4}$$

where η is the **normalizer** that ensures the probability sums to one, given by

$$\eta = \sum_{i=1}^{20} P(z_{t+1}|x_{t+1} = i)P(x_{t+1} = i|u_t)$$

As you can image, the probabilities $P(x_{t+1} = k|u_t, z_{t+1})$ over all the cells k at time $t+1$ constitute $bel(t+1)$.

A trick for the implementation of Eq. (4) is the so-called normalizer trick, which can be elaborated in three steps:

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

$$\eta = \sum_{k=1}^{20} \overline{P(x_{t+1} = k|u_t, z_{t+1})}$$

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

Please try to convince yourself that the above three steps do the exactly same thing as in Eq. (4).

Please download the template C++ code from UTS online.

2 How to Compile and Run

cd to the the bayes_filter directory

Process the following commands in terminal

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

3 Questions

The possible locations of the robot is preserved with a *vector* in the provided C++ code. So the probability at position k is actually saved in the *vector* indexed by $k - 1$. (The indices of the vector start from 0.)

3.1 Prior Belief

Suppose our prior knowledge about the **initial location** of the robot is as follows:

- We have a uniform distribution as our initial belief.

Q1. Compute the probability of $P(x_0 = k)$ for any $k \in \{1, \dots, 20\}$.
Q2. Initialise the vector of prior belief in the code.

3.2 Motion Model

Q3. Complete the function `motionProb(...)`:

- Inputs: x_{t+1}, x_t and u_t (two **potential** locations and the control input)
- Output: $P(x_{t+1}|x_t, u_t)$

3.3 Observation Model

Q4. Complete the function `obsProb(...)`:

- Inputs: z_t and x_t (we also know the exact location of the doors)
- Output: $P(z_t|x_t)$

3.4 Discrete Bayes Filter

Q5. Complete the function `bayesFilter(...)` based on Equation (3) and (4). Use `motionProb(...)` and `obsProb(...)` to compute the motion and observation probabilities, respectively.

Q6. Compute the belief at time $t = 1$ to $t = 4$ after processing the following sequence of control inputs and observations:

$$(u_0 = 0, z_1 = 1) \quad (u_1 = 1, z_2 = 0) \quad (u_2 = 1, z_3 = 1) \quad (u_3 = 1, z_4 = 1)$$

Q7. Explain your results in **Q6**.

Q8. Why do we need to observe the doors (or any other landmark)? What would happen if we stopped using our sensor?