Report for assignment 2

Jingmo Bai

February 17, 2023

1 Statement

I did the peer review with Robin Öhrnberg, whose student ID is “jur11roh”. I also had some discussions with him during the implementation.
After the peer review of the implementation, I modified the sensor reading part. My peer commented that I did not handle the case of no reading, and the high accuracy is abnormal. I realized that I use the true state position as the observation result, that’s why I get high accuracy. So I changed it to the result of the sensor reading and also consider the case of no reading.

2 Summary of the task

Our task in this assignment is to track a robot’s location. The scenario is the robot is in an empty room without any landmarks which can be represented by a rectangular grid. The robot moves according to some known rules and strategies.
All the information we can acquire is from a noisy sensor called the observer. It will give approximation locations for the true location, the directly surrounding fields, or the secondary surrounding fields, all have some certain probability respectively. In addition, there is a probability for the sensor to produce ”nothing”.
To fulfil the task, we use forward filtering based on an HMM, hidden Markov model. We are given several models which implement the robot movement, observation, and state encoding.

3 Brief overview of the implementation

In my implementation, there are basically four parts.
First, the simulation of the robot’s movement, to get a ground truth value of its location. Second, the simulation of the sensor reading for the HMM filter. Third, the implementation of the HMM-based forward filtering. Forth, a loop over moving the robot, obtaining the sensor reading, and updating the estimate position with the forward algorithm based on the sensor reading.

4 Explanation of the models

The transition model contains the strategies of the robot to take the next move. The input is the current state of the robot, it will calculate the probability according to the robot moving strategy and finally give a transition matrix with the probability of each grid that the robot can move to as the output.
The observation model contains the rules of the sensor to give the readings according to the current state. It will give vectors of the observation matrices for each possible sensor reading.
The state model gives several methods to do transformations between state, position, pose, and reading.
By pressing the "show transition" button, we can see the probabilities of moving from each possible current state to all the other possible states in the visualisation of the transition model.
By pressing the "show sensor" button, we can see the probabilities for each position to be reported by the sensor. Especially, if we press the "show sensor" button before initializing the filter, we can see the probabilities of each position on which the sensor may report "nothing", we can see that the probability is higher around the edge of the grid, and it's lower in the centre of the grid.
When we run the filtering process using the GUI, we can see the simulated true position as black, the sensed position as cyan, and the guessed position as red. If the sensor report "nothing", then there is no cyan block. The probabilities of the guessed position are shown in the form of a heat map.

5 Discussion of results

To evaluate the performance of the algorithm, I use a grid size of 6x8 and run it until 100 steps.
The performance of the estimation of the HMM filter achieves an average accuracy of 34% and an average error of 1.778 by using the Manhattan distance. Compared to pure guess by selecting the states randomly, pure guess can only achieve 2% accuracy. Compared to only the sensor readings by removing the HMM filter, the sensor can achieve about 10% accuracy.
In summary, HMM-based forward filtering is much better than just making random guesses or using the sensor reading.

6 Summary of the article

The paper presents an algorithm for robot localization, called Monte Carlo Localization (MCL). It is a version of Markov localization. Some of the advantages compared to previous methods like Markov localization using grids are that MCL requires less memory and computation, it is more accurate, and it is much easier to implement.
MCL applies the importance re-sampling method for approximating posterior distributions, in a way that places computation where needed. The key idea is to represent the posterior belief by a set of N-weighted, random samples or particles. In analogy with the previous Markov localization approach, MCL proceeds in two phases: Robot motion and Sensor readings. When the robot moves, new samples are generated by randomly drawing a sample from the previously

computed sample set, with likelihood determined by their weighting factor. In sensor reading, the sample set is re-weighted in a way that implements Bayes rule in Markov localization.
After the evaluation of MCL, two primary results are obtained. First, MCL yields significantly more accurate localization results, while consuming less memory and computational resources. Second, adaptive sampling performs equally well as MCL with fixed sample sets.

7 Relation between my work and the work in the article

I believe that the question, if HMM approach is suitable for solving the problem of robot localization, depends on the environment of localization. For the scale of our assignment, HMM can complete the task. However, if we want to do it in a real-world environment, HMM could fail, and MCL is more suitable because it requires less memory and computation.

A Comment received from Robin Öhrnberg

I've looked through your code. It's very concise and elegant. The robot moves as it should only taking legal moves. The sensor seems to work as intended, although I can't see how you handle the case of no reading. The filter also seems to work very well considering the high accuracy rate. Other than that I don't have any remarks as your code works and my mind don't.