Localization with Bayesian Filter

Group 9 - Xiaomi Redmi 5A (Android 7.1.2, API 25)

Sonnya Dellarosa
4783344

Pranjal Rajput
4788087

I. Introduction

In this assignment, we have to develop an Android application that provides the location of the user within 16 cells on the fifth floor of the EWI building.

II. Method

A. Data collection

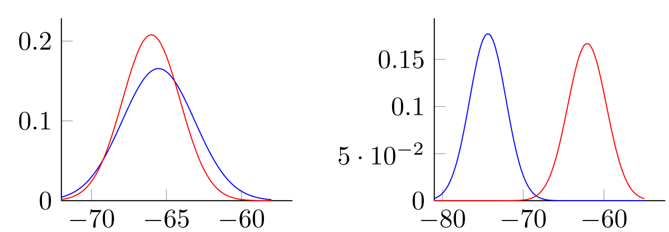
We collected the RSSI information of the available access points in each of the 16 cells mapped in the provided floor plan of the 5th floor of the EWI building. A total of 100 samples were collected for each cell, with a time interval of 300 ms between each Wi-Fi scan. We collected the data in two different scenarios, when the floor was empty (scenario 1) and when it was crowded (scenario 2).

B. Data processing

We performed an offline analysis of the collected data to filter out all the access points besides *eduroam* or *tudelft-dastud*, as these two can be considered permanent access points. We, then, split the collected samples for training (75 samples) and testing purposes (25 samples). The testing samples are then put into groups of 5 such that there are 5 tests of 5 iterations. We trimmed the outliers in the training data by calculating their z-score values and removing those with a value above the threshold, which we set as 2.

C. Radio map

Our offline analysis indicated that using a Gaussian distribution provides a higher accuracy than using a histogram. Therefore, we decided to represent the probability mass function (PMF) of the RSSI values using a Gaussian distribution.



(a) PMF of cells 3 and 4

(b) PMF of cells 1 and 5

Fig. 1: PMF example of similar and different cells of one AP

III. Evaluation

We tested the app in 8 cells, from cell 9 to cell 16. We start by pressing the *Initial Belief* button, that initially sets an equal probability of $1/n$ for each cell. Then we select a random cell to test, and press the *Locate Me* button for one iteration, until the probability converges.

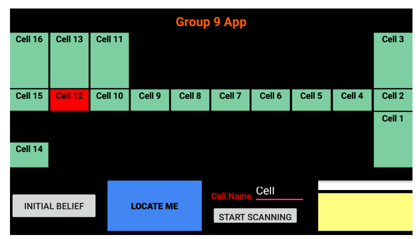


Fig. 2: Screenshot of the application

IV. Discussion

While testing the application, we noticed the following points.

- **Similar PMFs**: Where the cells are small and next to each other, the PMF of the cells might be very similar and indistinguishable such that the application is not able to detect the correct cell. An example of this case can be seen in Figure 1 (a).
- **Fading effect**: When the floor is crowded, the probability of detecting the correct cell is observed to be a lot less accurate than when the floor is empty. Our data from scenario 1 gives an accuracy of 0.99 during testing while our data from the scenario 2 produces a much lower accuracy of 0.79. This may be due to multi-path fading, i.e. absorption effect, which causes the RSSI values to fluctuate.
- **Zero probability**: At first, we implemented our application such that whenever the PMF does not exist for the scanned AP, then the perception model probability used is 0. However, this means that if the first scan wrongly puts a probability of 0 to the true cell, this value will be in the prior probability for the next iteration and will stay at 0 in consequent iterations. The application would never be able to converge to the true cell. For this reason, we changed our implementation where we instead of 0 we use a small value ϵ. After trying different values, $\epsilon = 0.01$ seems to provide the best result.

APPENDIX

The following is the confusion matrix of samples in both scenarios combined, therefore a total of 150 training samples and 50 testing samples, tested for 5 iterations.