Smart Phone Sensing - Report 1

Group: `7`, SmartPhone: `OnePlus 6T`, Android Ver: `10`, Git Branch: `main`

Minas Melas
NetID: mmelas
Student No.: 5397324
Email: M.Melas@student.tudelft.nl
Labour: Localization

Nishad Mandlik
NetID: nmandlik
Student No.: 5242541
Email: N.G.Mandlik@student.tudelft.nl
Labour: Activity Recognition

***Abstract*—Localization and Motion sensing are two topics with many implementations that have and are being developed throughout the years. One of the techniques that have been developed is the analysis of different RSSI signals gathered from nearby Wi-Fi Access Points. As for the motion sensing, the analysis of signals gathered by the gyrocopter and the accelerometer can produce results with great accuracy. Of course, the accuracy of both these always depends on the technique used and on the filtering of the received signals.**

We used k-fold Cross-Validation kNN for both these implementations, by using different features and filtering for each one of them as discussed in the next sections. Then, we created the confusion matrices for some interesting tests that we performed and we analyzed the results.

I. Method

A. Wi-Fi Localization

For the Wi-Fi localization, we did 400 scans (during the day) for each room with a delay of 200 ms between them and used the average of three consecutive scans as a feature. The signal strength is a very good (but not always consistent) indicator of the distance from the access points and the average was taken becuase there is a possibility that the device might skip a signal even if it is in close distance. We did several measurements on the times between the Wi-Fi signals and we found out that the 200 ms, which the device was capable of supporting, was a good choice according to the accuracy of the results.

We used six cells each indicating a location inside the house (Bedroom1, Bedroom2, Kitchen, Balcony, Toilet, Living Room). An estimate of the distances between these rooms (the distances between the average centers of the recorded signals) are depicted in Table I and the layout of the house in Figure 2.

After some research on distance metrics, we found that Euclidean distance was the best choice. Euclidean distance is the shortest path from the source to the destination, which is useful for the Wi-Fi localization, since the signal is dependent on this distance as well. The value of k is dynamically selected by using k-fold Cross Validation kNN, as described in section II.

B. Activity Recognition

We observed that the in between the activities of sitting still and walking, there is a significant variation in the accelerometer signals. Thus, it would be possible to differentiate between these activities by using the accelerometer data alone. The mean of the accelerometer values is similar for both activities, and hence, is not a good distinguishing feature. The nature of oscillations varies. The walking produces oscillations of higher magnitude. Thus, we choose the standard deviation and range (max. - min. value) as the features of the signal. We selected push-ups as the third activity, since it gives feature values similar to those of the walking activity. In order to make differentiation possible, we decided to use the data from the gyroscope as well. We characterise this data using the same features as that for the accelerometer. In the accelerometer output, the major contribution is due to the gravitational force of the earth. Thus, the individual x-, y- and z- components are highly influenced by the orientation of the smartphone. Hence, we use the resultant of the x-, y- and z- components. The gyroscope, on the other hand is not affected by gravity or any other external force. Hence, we decided to use its each component as an individual parameter. By visual inspection of the signals (Figure 1), we observed that no two activities show similar features for all 4 signals. Thus, by including the gyroscope data (i.e., adding more dimensions), we are able to achieve a considerable separation between data points of different classes, which would not be the case, if only accelerometer data was used. So finally, we have 4 components, each having 2 features, which results in an 8-dimensional data space. The cross-validation process for computing k follows similar logic as that for WiFi-Localization.

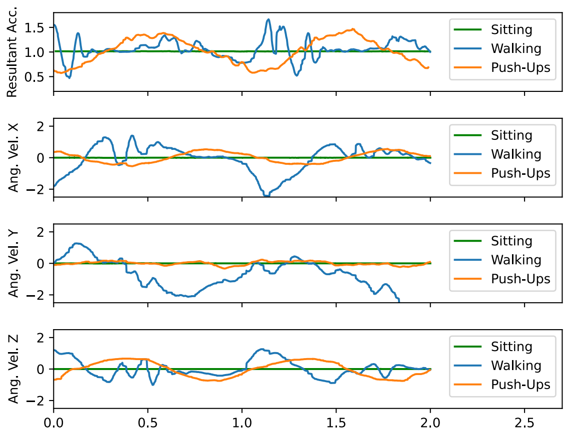

Fig. 1: Comparison of Accelerometer (magnitude of resultant) and Gyroscope (x-, y- and z- components) readings during various activities

II. Evaluation Setup

A. Wi-Fi Localization

In total we gathered 804 data from the scans of the 6 rooms. We used 4-folds for the k-fold validation. In each run, we shuffled the data randomly in order to remove any correlation that might exist due to the consecutive scans. Consequently, we kept 20% of the total data (161 samples) for testing and the rest (643 samples) were used for training. Out of these 643 samples, each fold uses 1/4 of the set for training and the rest for testing. The accuracy is then computed for each fold, by calculating the kNN and the corresponding successful results. This is repeated for different k values ($1 \leq k \leq 13$) in order to find the best k value. After finding the best k value (according to the accuracy results) we test the trained set with the test set that we kept in the start. If this accuracy is not good enough (less than 85% accuracy) then we rerun the kNN k-fold Cross-Validation (up to 5 times) which will shuffle the data differently and thus produce a different accuracy. In the end, the best k value will be saved and will be used with the whole (804 samples) data for locating the user whenever the button "IDENTIFY POSITION AND ACTIVITY" is pressed.

We performed two different tests, the results of which are shown in Table II, one for having the mobile unmoved in a position while gathering the training and test data and one where we were moving in a 2x2 location inside the cell while scanning the values.

The reason for this differentiation is that RSSI values are very sensitive to change in position. Thus, keeping the mobile in the same position (both for training and testing) is expected to have better accuracy results, since the values both in the training and in the testing will be very close to each other. That's a possible reason that $k = 1$ is the best value for the static position, since there will be much difference between the different

RSSI values gathered by the different cells (and thus the features used in the kNN). However, keeping it in the same position will not have a good generalization of the whole cell and does not produce a good accuracy when trying to locate different positions around the cell. On the other hand, gathering data while moving produced a lower accuracy result but had a better generalization around the cell. The dynamic (moving) version produced a best value of $k = 7$.

TABLE I: Distances (in m) between Rooms

	Living Room	Kitchen	Toilet	Bedroom1	Bedroom2	Balcony
Living Room	0	3	5	6	6	8
Kitchen	3	0	3	5	4	7
Toilet	5	3	0	4	3	6
Bedroom1	6	5	4	0	3	2
Bedroom2	6	4	3	3	0	3
Balcony	8	7	6	2	3	0

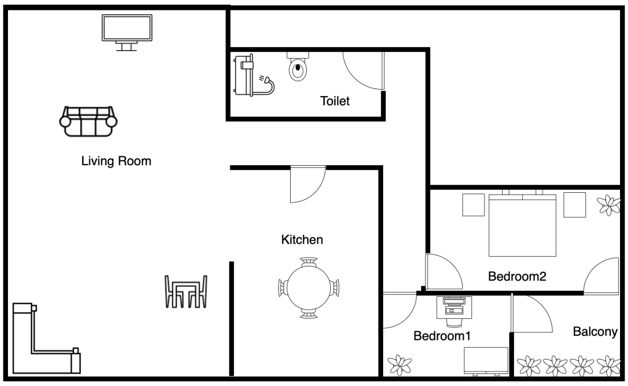

Fig. 2: Home layout

B. Activity Recognition

For the experiments, the training and testing was performed on the same device. The smartphone was positioned in the front pant-pocket. The training process involves collecting sensor data for 45 seconds, and computing features for overlapping windows having duration of 2 seconds. We select a 50% overlap between successive windows. Since the sensor data is read whenever the sensor generates an event, the number of samples in a each window is different. But since we are only interested in the standard deviation and range of values in a window, we treat one window as a single sample. For the selected duration, window size and overlap, this results in 44 samples for each activity (132 samples in total). The testing was carried out using a total of 27 samples consisting of 9 samples for sitting, 10 for walking and 8 for push-ups.
With the cross-validation approach, as mentioned earlier, the best value of k was computed as 1. This implies that the outcome of a measurement is determined by only the closest data-point from the training set. Thus, it can be said that the data-points of different classes are sufficiently separated in the feature-space. Hence, a single nearest data-point is sufficient to estimate the result without ambiguity.

III. Results

The confusion matrices for localization and activity detection are shown in Table II and Table III respectively.
The accuracy of the results was 91.92% and 79.5% for static and dynamic movement during gathering the signals respectively for the localization. The accuracy for activity detection was 96.30%. The GUI of the app can be seen on Figure 3.

Fig. 3: App home screen (top left), and screens for for Localization Training (top right), Activity Training (bottom left), and Recognition of Both (bottom right)

TABLE III: Confusion Matrix for Activity Recognition

Actual \ Predicted	Sitting	Walking	Push-Ups
Sitting	9	0	0
Walking	0	10	0
Push-Ups	0	1	7

TABLE II: Confusion Matrices for static (left) and dynamic (right) training and testing of localization

Actual \ Predicted	Living Room	Kitchen	Toilet	Bedroom 1	Bedroom 2	Balcony
Living Room	25	0	0	0	0	0
Kitchen	0	25	0	0	0	1
Toilet	4	1	22	0	0	1
Bedroom 1	0	0	0	24	0	0
Bedroom 2	0	1	0	1	27	3
Balcony	0	0	0	1	0	25

Actual \ Predicted	Living Room	Kitchen	Toilet	Bedroom 1	Bedroom 2	Balcony
Living Room	13	6	3	0	0	0
Kitchen	2	24	5	0	0	0
Toilet	0	0	27	0	1	0
Bedroom 1	0	0	0	32	0	0
Bedroom 2	0	0	1	0	10	14
Balcony	0	0	1	0	0	22