Mobile Health and Activity Monitoring Subtask3 Report

Enjie Cao, Yifan Zhou, Zhichao Sun

In this subtask, we developed an algorithm using previously recorded datasets with wearable devices to predict four different aspects of the user's physical activity: activity type, path taken, step count, and smartwatch location on body. Our algorithm was designed to predict each of these four aspects independently without results of other aspects as inputs, preventing errors from accumulating and improving the overall accuracy of the algorithm.

1 Signal Pre-processing

We only use sensor data from wristband accelerometer, wristband gyroscope and phone magnetometer which is always included in all data, thus ensures the validity of our algorithm.

First, as all three axes of the accelerometer sensor may contain gravitational acceleration, we split the gravitational acceleration with a low-pass filter around 0.2Hz and subtracted it from raw data. Second, according to [1], people walking frequency is between 1Hz and 3Hz. As we also have running and cycling in the dataset, we expanded the upper limit to 5Hz, filtering the signal from wristband sensors using a low-pass filter of 5Hz and a high-pass filter of 1Hz successively. The results of pre-processing are shown in Figure 1.

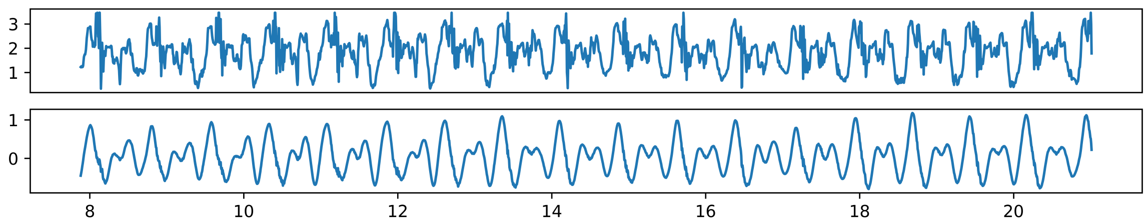

Figure 1: Accelerometer magnitude data before pre-processing (Upper) and after pre-processing (Lower)

2 Step Counting

Using the magnitude of 3-axis acceleration data is a common technique in wearable devices for step counting[2]. Our algorithm consists of following steps:

1. Windowing and Detecting Non-Moving Phase: To distinguish non-moving phases from moving phase, the signal over the whole trace is divided into non-overlapping 1-second windows, and for each window, a threshold of magnitude of acceleration is set to 0.04 based on trial-and-error.
2. Zeroing and Normalization: For non-moving phases, we aim to reduce its interference on later peak detection, so the acceleration magnitude during non-moving phases is set to zero. For moving phases, we aimed to count steps regardless of the activity type and its associated scale in acceleration. For this purpose, the acceleration magnitude is normalized into a range of [-0.5, 0.5] independently.
3. Peak Detection: A *findpeak* function is applied to the processed signal of the entire trace and identifies the local maxima. The number of peaks detected is then counted as the number of steps taken.

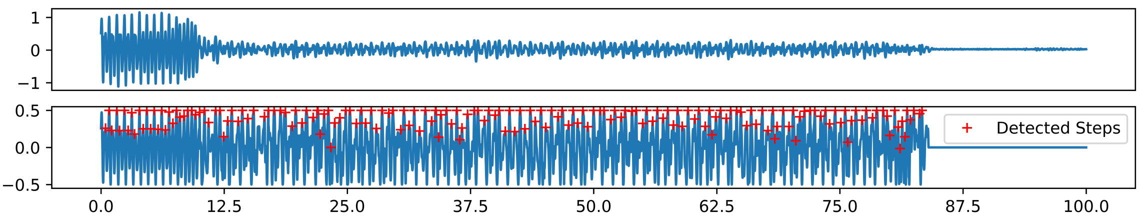

Figure 2: Accelerometer magnitude data after pre-processing (Upper) and after step counting (Lower)

3 Location, Path, Activity Detection

3.1 Feature Extraction

For location and activity detection, accelerator and gyroscope data were divided into sliding windows. In each window, several frequency and time domain features were extracted to reflect the impact of each step or movement on sensor measurements, as proposed by Vahdatpour et al.[3], Debache et al.[4] and Zhang et al.[5].

3.1.1 Time domain features

- $acc\_mag\_mean, acc\_mag\_std, acc\_mag\_skewness, acc\_mag\_kurtosis$: Mean, standard deviation, skewness, and kurtosis of the acceleration magnitude
- A, A_m: maximum acceleration range, maximum acceleration mean among 3 axes
- B, B_m: ratio of the middle and maximum acceleration range, ratio of the middle and maximum acceleration mean among 3 axes
- C, C_m: ratio of the minimum and maximum acceleration range, ratio of the minimum and maximum acceleration mean among 3 axes
- $Q_1, Q_2, \ldots, Q_12$: the [0, 5, 10, 20, 30, 40, 50, 60, 70, 80, 90, 95, 100] percentiles of acceleration magnitude
- $range$: acceleration magnitude range
- rms: root-mean-square of acceleration magnitude
- $zero\_crossing$: the number of times the signal crossed the mean

Feature A reflects motion range, features B, C reflect degree of freedom in movement and dependence of axes. Inspired by this idea, we further extracted Am, Bm, Cm with mean instead of range, and features of gyroscope data G, H, I, G_m, H_m, I_m.

3.1.2 Frequency domain features

- E, H, c, b: Energy, Entropy, Centroid, Bandwidth
- f_1, f_2: dominant and second dominant frequency of acceleration magnitude
- p_1, p_2: power of $p1, p2$

3.2 Location Detection

3.2.1 Training Data

First, we split the given training dataset into training set (220 traces) and validation set (50 traces) in order to see validation accuracy variation during training. Second, the raw wristband accelerometer and wristband gyroscope data of three axes are preprocessed (section 1) and segmented along the time axis into windows of 10 seconds with an 50% overlap [6, 7]. Thirdly, we removed all non-moving windows since they can't offer distinctiveness on location information. Lastly, the following features are calculated from each moving segmented data window: $acc\_mag\_mean, acc\_mag\_std, A, B, C, Am, Bm, Cm, G, H, I, E, H, f_1, p_1$. Additionally, in all training windows, we only use 70% windows to do training, the rest 30% windows used to validate detection accuracy on each window (which is different to the validation accuracy of the whole trace we mentioned before).

3.2.2 Model Selection

We decided XGBoost[8] for this classification task due to its high performance, its ensemble of decision trees, and its fast training speed compared to other decision trees. And we used grid search to tune the hyperparameters of the model.

3.2.3 Results

The validation accuracy on segmented window data is 95.84%. For the prediction of a trace, we use the mode of all window predictions, whose validation accuracy is 95.35%.

3.3 Path index Detection

3.3.1 Training Data

First, we split the given training dataset into training set (80%) and validation set (20%) in order to see validation accuracy variation during training. Second, we calculate the magnitude of smartphone magnetometer among 3 axes. And since different traces have different length, we downsampled all traces to 1000 data points.

3.3.2 Model Selection

We decided CNN [9] for this classification task because CNN can capture spatial relationships, which captures the features of change of magnetic force for different paths.

3.3.3 Results

The accuracy of our model on validation set is 75.47%.

3.4 Activity Detection

3.4.1 Training Data

First, we calculated the magnitude of wristband acceleration across all three axes, are pre-processed and segmented along the time axis into windows of 3 seconds with an 50% overlap [4]. Second, we picked 64 traces out of 270 training traces and labelled the activity of all segmented windows by hand, distribution of labelled window are shown in Table 1.

Table 1: Distribution of windowed activity labels

Activity	Number of windows
Stand	2645
Walk	19755
Run	1470
Cycle	1344

Third, the following features are calculated from each labelled segmented data window: acc_mag_mean, acc_mag_std, $acc\_mag\_kurtosis$, $acc\_mag\_skewness$, $rms, zero\_crossing$, $range$, $Q_1, Q_2, \ldots, Q_12$, E, H ,c, b, f_1. Additionally, in all training windows, we only use 80% windows to do training, the rest 20% windows used to validate detection accuracy on each window.

3.4.2 Model Selection

We decided XGBoost [8] for this classification task because our training data is very imbalance, and XGBoost handles this problem very well.

3.4.3 Results

The validation accuracy on segmented window data is 94.90%. For the prediction of each activity of a trace, we use a sliding window of 60s with 33% overlap, and use the mode of each sliding window as a valid activity. We also tested our model on the rest 206 traces without labelled windows, and the accuracy for each activity are shown in Table 2.

Table 2: Validation accuracy of different activities

Activity	Accuracy
Stand	65.40%
Walk	96.58%
Run	89.73%
Cycle	93.92%

4 Conclusion

In conclusion, with the extracted features above, our model achieves considerate performance on location and activity detection. However, more improvement is still needed on path index detection, as the magnetometer data can be easily influenced by subtle changes in the ambient magnetic field, such like nearby electronic devices.

References

[1] A. Pachi and Tianjian Ji. Frequency and velocity of people walking. 2005.

[2] Agata Brajdic and Robert Harle. Walk detection and step counting on unconstrained smartphones. In Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing, 2013.

[3] Alireza Vahdatpour, Navid Amini, Wenyao Xu, and Majid Sarrafzadeh. Accelerometer-based on-body sensor localization for health and medical monitoring applications. 2011.

[4] Isaac Debache, Lorène Jeantet, Damien Chevallier, Audrey Bergouignan, and Cédric Sueur. A Lean and Performant Hierarchical Model for Human Activity Recognition Using Body-Mounted Sensors. 2020.

[5] Shaoyan Zhang, Alex V. Rowlands, Peter Murray, and Tina L. Hurst. Physical Activity Classification Using the GENEA Wrist-Worn Accelerometer. 2012.

[6] Ozlem Durmaz Incel. Analysis of Movement, Orientation and Rotation-Based Sensing for Phone Placement Recognition. 2015.

[7] Vu Ngoc Thanh Sang, Shiro Yano, and Toshiyuki Kondo. On-Body Sensor Positions Hierarchical Classification. 2018.

[8] Tianqi Chen and Carlos Guestrin. XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016.

[9] Jiuxiang Gu, Zhenhua Wang, Jason Kuen, Lianyang Ma, Amir Shahroudy, Bing Shuai, Ting Liu, Xingxing Wang, Gang Wang, Jianfei Cai, and Tsuhan Chen. Recent advances in convolutional neural networks. 2018.