Activity Recognition with Phone Sensors

Digital Signal Processing

R Arvind, Anubhav Mishra

International Institute of Information Technology, Bangalore, India

Abstract

With the advent of smart gadgets, human activity recognition (HAR) has gained a lot of attention due to the number of applications it has. Motion sensor embedded smartphones give insight into the user's physical activities and allow the user to make more informed and healthier choices. Detecting physical activities like running, walking, sleeping, climbing and other information like the number of calories burnt, etc is the result of research in the field of HAR. In this project, we are trying to predict the activity performed by the user using the smartphone sensors by applying Machine Learning techniques to the data obtained.

Introduction

Activity recognition is the task of recognizing the actions of individuals, based on the limited information from sensors on and around the individual.

In this project, we will focus on the use of data acquired from the user's smartphone. The next question is what all information can be gathered from a smartphone? Our smartphones are embedded with a number of sensors each having its own set of the application. Some of them are Proximity Sensor, Magnetic Sensor, Gyroscope, Accelerometer, Ambient Light Sensor, Barometer, etc. The sensors we are using for collecting data are the Accelerometer and the Gyroscope.

An accelerometer is a sensor that can measure the force acting upon it, be it from physical acceleration or from the Earth's gravity. Gyroscope tells the measure of the angular velocity of the body. It can be used to detect the motion of the phone along its axis.

Data Collection

The experiment was carried out in a group of 4 people. Each person performed all four activities (walking, climbing, running, jumping) with smartphones in their hands. The orientation of the phone was left up to the choice of the person performing the task. This was done in order to introduce variety into the data set and to avoid bias towards one type of orientation. All the data was collected on different smartphone devices and then combined together. --

The data was collected from the smartphone using an android application called "*Sensor Record*". The app uses the embedded accelerometer and gyroscope to capture the data. It also allows us to set the sampling rate of the data.

The app captures 3-axial linear acceleration and 3-axial angular velocity and provides a CSV file with the data and the corresponding timestamp.

Pre-Processing

The data rate of the sensors was set at 100Hz i.e the application gave 100 readings every one second. This data was stored in a CSV file and then processed using python.

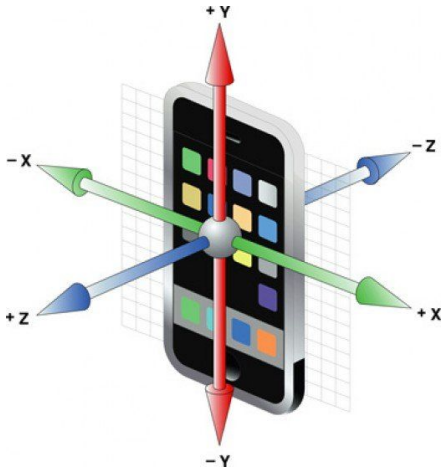

Fig 1: The three-axis of accelerometer

In order to remove outliers from the data, the first 100 readings and the last 100 readings were removed. This will eliminate human error from the data. The number of data points was trimmed down to the nearest multiple of hundred for the ease of making windows.

The data was further split into windows of 100 readings each(100 samples = 1 second of data) with a 50% overlap between two consecutive windows. Also, each reading is an array of 6 values (Acc x,y,z and Gyro x,y,z).

Table 1: Labels assigned to the activities

Activity	Label
Walking	1
Running	2
Jumping	3
Climbing	4

Each window is assigned a label depending on the activity which was performed. For this project, we used four different types of activity as mentioned in Table 1.

Data Exploration

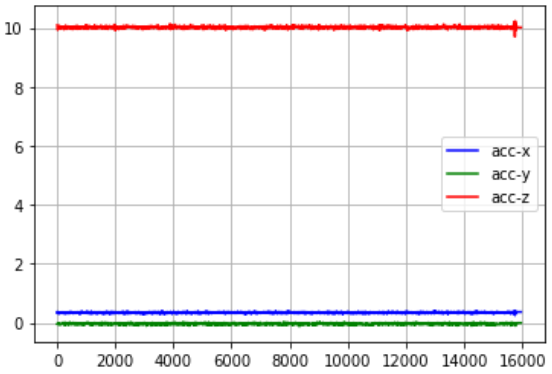

Fig 2: Acceleration vs Time(ms) when Stationary

From the plot of Acceleration vs time for stationary phone, we can see that all the values are almost constant. The z acceleration is around 10 m/s^2, which is due to gravity.

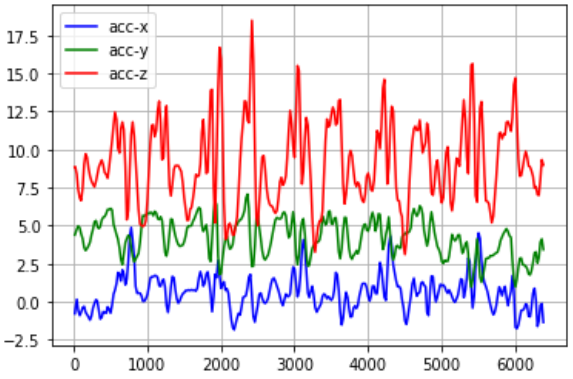

Fig 3: Acceleration vs Time(ms) when Walking

We are able to clearly differentiate between the plot for walking and jumping.

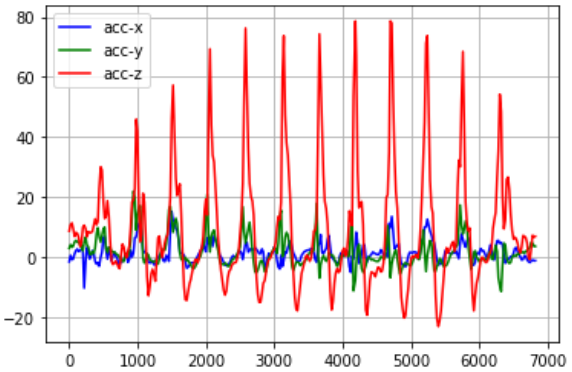

Fig 4: Acceleration vs time(ms) for Jumping

While jumping, the change of acceleration is the z-axis is dominant as compared to the acceleration in other axes. Whereas in the case of walking, there is roughly a change in all the axis. This is due to the fact that our hands also move slightly while walking. One more thing to note is that in case of jumping, the acceleration appears to be somewhat periodic due to the fact that the person was jumping periodically.

The plots of the gyroscope values are a bit more chaotic than that of the

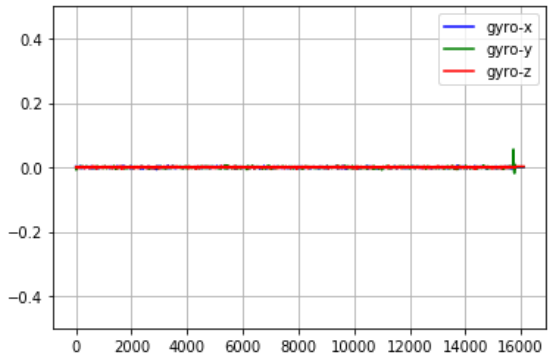

Fig 5: Angular Velocity vs time(ms) for Walking

The plot for the stationary state is as expected, there is no change in any axis.

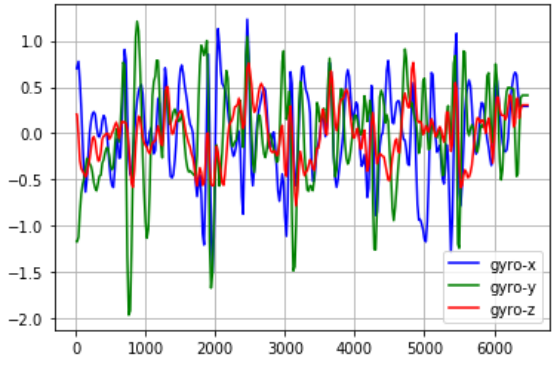

Fig 6: Angular Velocity vs time(ms) for Walking

But for the case of other activities, it is hard to infer anything from the plots of the gyroscope. These plots show the angular velocity of the phone about its axis. These values depend on how the phone was kept during data collection. If the phone were to be kept in the same orientation, then the gyroscope values will always be zero irrespective of the activity.

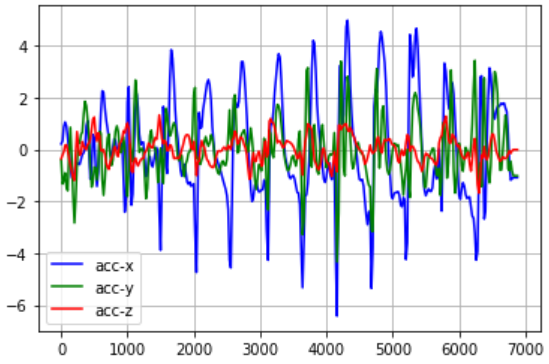

Fig 7: Angular Velocity vs time(ms) for Walking

These fluctuations in the plots show that the person kept moving is hand while performing the task which gave rise to these values.

Feature Extraction

In order to train our model, we need to get some meaningful values out of this data. Once the data was split into windows, we extracted multiple features from each window. Most of them are basic statistical quantities:

1. Mean :

$\sum x_i / n$

where x_i denotes the values of each column in the window and n is the size of the window.

2. Standard Deviation:

$SD = \sqrt{(\Sigma(x_i - \underline{\mathbf{x}})/n)}$

$\underline{\mathbf{x}}$ = mean of the column in the window

x_i = each data in that particular column in the window

n = length of the window

4. Median Absolute Deviation:

$MAD = median(|x_i - median(\mathbf{x})|)$

x_i is the element of the column in the window.

x is the column under consideration in the window.

5. Minimum Value:

The minimum value in the window

6. Maximum Value :

The maximum value in the window

7. Average derivative:

The derivative of each point in the window is calculated and the average is taken.

Building the Model

Given the nature of the problem, it is clear that we need to classify the output. Since SVM is one of the most robust and accurate algorithms among the other classification algorithms, we decided to use SVM for this project.

The main objective in SVM is to find the optimal hyperplane to correctly classify between data points of different classes.

If the data we are working with is not linearly separable, SVM employs a technique called Kernel Trick. This method is able to map our nonlinear separable data into a higher-dimensional space, making our data linearly separable.

The SVM that we applied uses RBF (Radial Basis Function) as the kernel. All the other parameters were set to default.

The dataset was split into a 7:3 ratio, 7 parts for training and 3 parts for testing the accuracy of the model.
This task was repeated multiple times, each time the testing and the training datasets were shuffled and fed into the model.

Observations

The collected data is time-series data. The sampling rate of the sensors was set at 100Hz i.e the app gave 100 readings every one second. This data was stored in a CSV file and then processed using python.
Both the sensors are tri-axial. Therefore there we 6 readings collected, acceleration in x,y and z-axis and angular velocity in x,y, and z-axis.

On training the model, the accuracy turns out to be **91.7%.** This is the average score over multiple train/test cycle.

Table 2: Confusion Matrix (Actual vs Predicted)

	Climbing	Walking	Jumping	Running
Climbing	46	15	0	0
Walking	3	80	0	0
Jumping	0	0	29	0
Running	0	0	0	45

A confusion matrix shows the breakdown of correct and incorrect classifications for each class. In the above table, the columns are the actual values and the rows are the predicted value.
From the confusion matrix, we can see that the model has difficulties differentiating between walking and climbing.

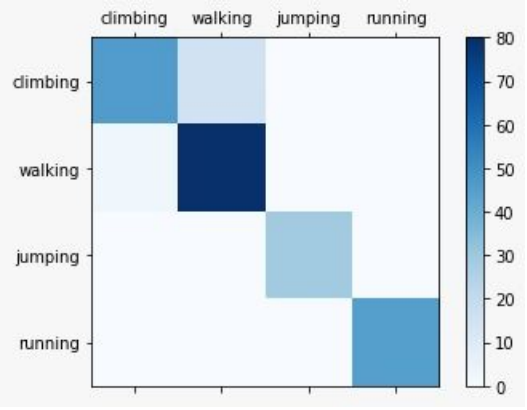

Fig 8: Confusion Matrix Plot (Actual vs predicted)

Conclusion

With the collected data, we were able to achieve **91.7%** accuracy.

1. Although we were able to achieve 91.7% accuracy, the model has problems differentiating between walking and climbing. The reason is that the features we extracted were mostly the basic ones like mean, standard deviation, etc.
There are more features like the energy, frequency components in fft which provides more information about the plot. But due to time constraints, we decided to use basic features.

2. We collected gyroscope values for all the activities, but most of the data collected didn't have much variation in the gyroscope data as we made sure not to move the phone much during data collection. Therefore the gyroscope doesn't contribute much to the predicted outcome.

Table 2: Confusion Matrix without Gyroscope

	Climbing	Walking	Jumping	Running
Climbing	40	19	0	0
Walking	3	84	0	0
Jumping	0	0	24	1
Running	0	0	0	50

We achieved 90.8% accuracy even without using the gyroscope values.

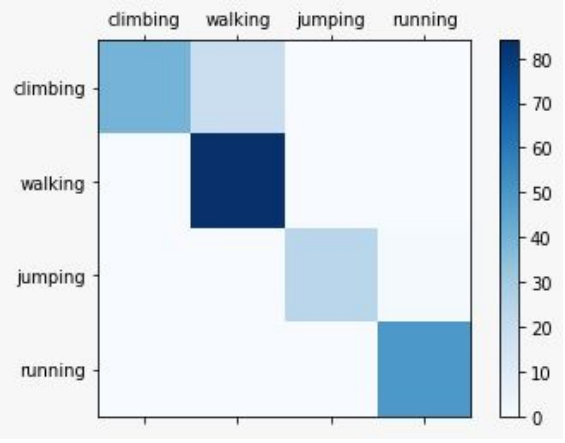

Fig 9: Confusion Matrix Without Gyroscope

We can see from the confusion matrix that there isn't much difference between that of output with gyroscope values and output without gyroscope values.
Change in angular velocity occurs when there is a change in the orientation of the smartphone. This allows us to detect particular hand movements such as waving or shaking which will be reflected in the gyroscope values. Hence, Considering the gyroscope values we can identify the change in orientation and take countermeasures for this movement.

3. Instead of considering separate features for each axis, we can find a feature that is a combination of all these features of multiple axes. For example, instead of the mean of acceleration in the x-direction alone, we can have a feature which is the sum of the means of accelerations in the x, y, and z directions. This helps in giving the same result for any change in the orientation of the smartphone.

References

- Alvina Anjum, Muhammad U. Ilyas. Activity recognition using smartphone sensors. IEEE 10th Consumer Communications and Networking Conference (CCNC), 2013.

- Human Activity Recognition using Smartphones DataSet