

CMPT 353 - PROJECT

Sensors, Noise, and Walking

Vafa Dehghan Saei - 301379021

Song Tung Nguyen - 301354423

Introduction

Our group decided to work on the “Sensors, Noise, and Walking” topic to be able to see what we could classify using accelerometer data. Our work included the full ETL (Extract - Transform - Load) process as well as using multiple machine learning techniques to classify our data to see which methods provided the best result for each problem. We wanted to know if we could determine the location of the phone, the owner of the phone, and whether there was an anomaly in the way the user walks. An example of an anomaly would be an imbalance between the left and right foot.

Data Gathering

Initially, we had planned on using a Fitbit smartwatch to determine the user's walking data. However, we soon learned that this device does not export raw walking data. It manipulates the data itself and displays it to the wearer in a more user-friendly way. Therefore, we had to use less-than-desirable methods to gather our data. We used an Android application called “Physics Toolbox Sensor Suite” on our phones to collect our data. This app collects many types of walking data, including g-force, linear accelerometer, gyroscope, and inclinometer. We decided on using only linear accelerometer data to train our machine learning classifiers. The reason we stuck with only one data type was that each person had to walk a ~15m path about six times for each data type. Therefore, we decided it would be better to focus on one type of data instead of delaying our work by collecting more. Furthermore, cleaning the accelerometer data proved to be more work than initially expected; therefore, it was better that we only stuck to one type of walking data. The cons of collecting more data types outweighed the pros, as we felt that linear accelerometer data was enough to classify our machine learning tools.

To gather our data, we strapped our phone onto both ankles, held it in both hands, and put it in the left and right pants pockets. We then started recording accelerometer data on the app, walked about 15 meters, and stopped recording. Each individual walked on relatively flat ground. Due to the nature of the pandemic, we could not meet up and walk the same path. Therefore, there may be small inconsistencies in the walking data. However, this is minuscule compared to the difference in pace, height, and gait of the participants. The application provided accelerometer data in a CSV file, with the data partitioned into a time, x-axis, y-axis, and z-axis column, with each of the xyz-axes containing acceleration data in m/s^2 . This format was handy to us as Pandas (our primary data manipulation tool), can work with CSV files natively.

Data Cleaning

Our primary job during the data cleaning phase was filtering the noise. The way in which we collected our data did not allow for any empty rows or duplicate values, so we did not have to filter those out.

As per the professor's advice, a mobile phone's sensor tends to have some bias and drifts that could affect the final result, with the solution being to stand still for a second at the start and the end. However, we did not take notice of this before we recorded the data; therefore, we decided to exclude the first 3 seconds as a part of the cleaning process. Clipping the data would eliminate both the drift and the strong fluctuation in the acceleration caused by the action of starting the recording and put it inside our pocket (for pocket data). To filter out the noise, we used a Butterworth filter, as was used by Maria Yousefian in her MSc thesis. To find out what

frequency we should set the filter at, we had to do some research. According to a research project from the Graduate School of Frontier Sciences at the University of Tokyo, they found that a second-order Butterworth filter with a cut-off frequency of 20Hz worked best at filtering out acceleration noise (Goto, et al., 2016). Another journal found that a 6 Hz Butterworth filter worked best (Fortune et al. 2014). Through trial and error, we discovered that we want to keep frequencies that are less than 0.02 (> 2 samples/second). The results turned out very good, almost all of the noise has been filtered out, and you can clearly see each step on the graph.

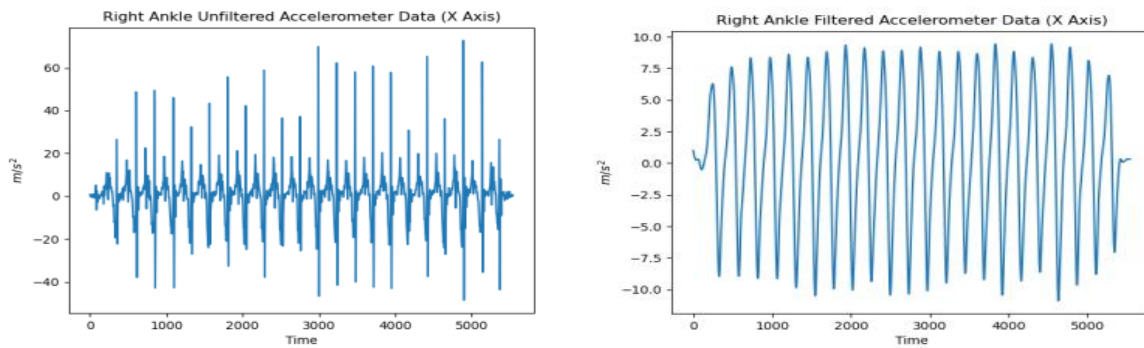


Figure 1 The raw data (left) and filtered using the Butterworth filter (right).

Analysis Design

After filtering out the noises using the Butterworth filter, we decided to perform the Fourier transform as recommended by the instructor to get the frequency of steps. We could then use the frequency of steps to analyze whether there is any difference between the frequency of steps between each person or detect any asymmetry between each person's left leg and right leg. Plotting the transformed data, we found that all the transformed data is relatively close to a normal distribution.

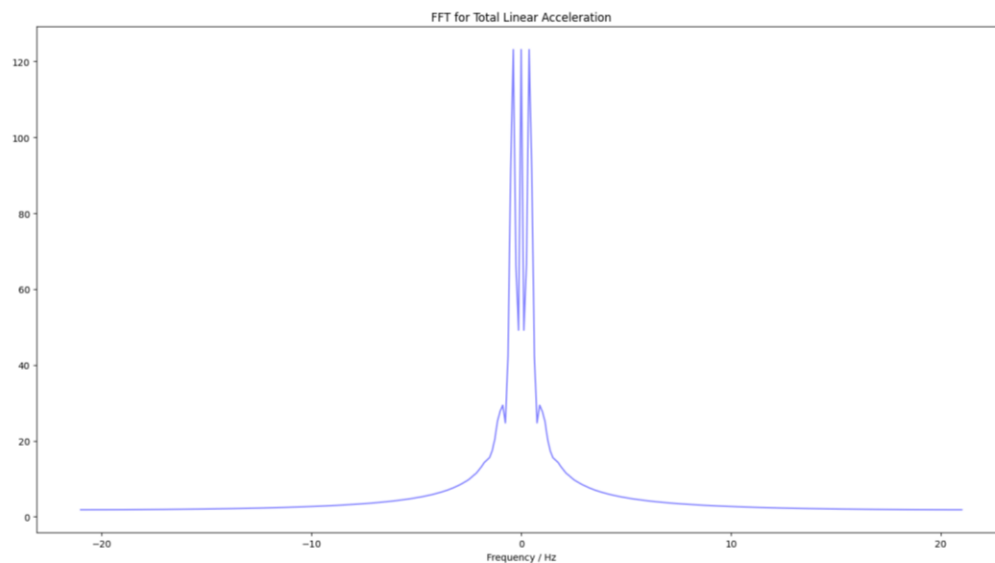


Figure 2 The total linear acceleration after applying Fourier transform

However, we could not rush to the conclusion that the transformed data follows the normal distribution just because it looks close to normal, so we decided to proceed with the following testing procedure:

1. Perform the normal test on the data to see if it is normal or not.
 - a. If the data is normal, perform the ANOVA test and Post-Hoc Tukey test to determine whether different placements affect the sensor reading or if the frequency of steps varies between each person's left and right leg.
 - b. If the data is not normal, we would perform the Mann-Whitney U-test to answer the same question.
2. If the result is inconclusive, what is the reason behind it?

As there are more than 4000 data points in each of the data files, it would be unreasonable to train the Machine Learning classifiers on the current set of data given the computing power we have. We decided to perform the training on the summary data (Figure 3) rather than the original data.

```
df = pd.DataFrame(summaries_person, columns = ['class',
'mean(ax)', 'std(ax)', 'min(ax)', '25th(ax)', '50th(ax)', '75th(ax)', 'max(ax)',
'mean(ay)', 'std(ay)', 'min(ay)', '25th(ay)', '50th(ay)', '75th(ay)', 'max(ay)',
'mean(az)', 'std(az)', 'min(az)', '25th(az)', '50th(az)', '75th(az)', 'max(az)',
'mean(aT)', 'std(aT)', 'min(aT)', '25th(aT)', '50th(aT)', '75th(aT)', 'max(aT)',
'mean(velocity)', 'std(velocity)', 'min(velocity)', '25th(velocity)', '50th(velocity)', '75th(velocity)', 'max(velocity)',
'mean(position)', 'std(position)', 'min(position)', '25th(position)', '50th(position)', '75th(position)', 'max(position)',
'mean(freq)'])
```

Figure 3 The rows represent: xyz axis, Euclidian total acceleration $\sqrt{ax^2 + ay^2 + az^2}$, velocity, and position, and the Fourier transform

Analysis Results

Inferential Statistics: For this section of the analysis, we were performing the analysis based on 3 different categories: The placement of the phone, owner of the phone, and the left & right side of the walker.

- **Normal Test:**

All datasets had $p < 0.05$, therefore we were able to reject the null hypothesis which means the data is not normally distributed. However, according to the instructor, if we have a big enough data set, we could assume that it is normal as stated by the Central Limit Theorem (CLT). Since each of the data set contains more than 4000 points, this effectively means that we would assume the data is normal and proceed to perform our analysis using the ANOVA test and Post-Hoc Tukey test rather than having to use Mann-Whitney U-test.

- **ANOVA Test:**

	Placement of phone	Owner of the phone	Side (L/R)
P-Value	1.7280703089280452e-06	1.799443995765991e-08	3.313281908823814e-09 (Song) 4.714252764063795e-10 (Vafa)

Based on the p-values above, we can conclude that there is a difference between the means within each group. Although the result of which side of the foot the data were recorded on allow us to conclude that there are differences between the frequency of steps of each foot, it is not enough to conclude the way we walk is asymmetrical, which will be explained in further detail in the Post Hoc Turkey analysis. Given the positive result from the ANOVA test, we decide to investigate further to see if the difference in the data of each foot means that there is an anomaly in the way we walk.

Multiple Comparison of Means - Tukey HSD, FWER=0.05						
group1	group2	meandiff	p-adj	lower	upper	reject
Vafa_left_ankle	Vafa_left_hand	-5.537	0.001	-8.4261	-2.6479	True
Vafa_left_ankle	Vafa_left_pocket	-4.2118	0.001	-7.1009	-1.3227	True
Vafa_left_ankle	Vafa_right_ankle	-1.0879	0.8885	-3.977	1.8012	False
Vafa_left_ankle	Vafa_right_hand	-5.5341	0.001	-8.4232	-2.645	True
Vafa_left_ankle	Vafa_right_pocket	-4.0864	0.001	-6.9755	-1.1973	True
Vafa_left_hand	Vafa_left_pocket	1.3252	0.7533	-1.5639	4.2143	False
Vafa_left_hand	Vafa_right_ankle	4.4491	0.001	1.56	7.3382	True
Vafa_left_hand	Vafa_right_hand	0.0029	0.9	-2.8862	2.892	False
Vafa_left_hand	Vafa_right_pocket	1.4506	0.6818	-1.4385	4.3397	False
Vafa_left_pocket	Vafa_right_ankle	3.1239	0.0252	0.2348	6.013	True
Vafa_left_pocket	Vafa_right_hand	-1.3223	0.7549	-4.2114	1.5668	False
Vafa_left_pocket	Vafa_right_pocket	0.1254	0.9	-2.7637	3.0145	False
Vafa_right_ankle	Vafa_right_hand	-4.4462	0.001	-7.3353	-1.5571	True
Vafa_right_ankle	Vafa_right_pocket	-2.9985	0.0367	-5.8876	-0.1094	True
Vafa_right_hand	Vafa_right_pocket	1.4477	0.6835	-1.4414	4.3368	False

Song_left_ankle	Song_left_hand	-4.4115	0.001	-7.4457	-1.3773	True
Song_left_ankle	Song_left_pocket	-2.7055	0.1124	-5.7397	0.3288	False
Song_left_ankle	Song_right_ankle	1.3181	0.7932	-1.7161	4.3523	False
Song_left_ankle	Song_right_hand	-4.2536	0.001	-7.2878	-1.2194	True
Song_left_ankle	Song_right_pocket	-3.1868	0.033	-6.221	-0.1526	True
Song_left_hand	Song_left_pocket	1.7061	0.5827	-1.3282	4.7403	False
Song_left_hand	Song_right_ankle	5.7296	0.001	2.6954	8.7638	True
Song_left_hand	Song_right_hand	0.1579	0.9	-2.8763	3.1921	False
Song_left_hand	Song_right_pocket	1.2247	0.8439	-1.8095	4.259	False
Song_left_pocket	Song_right_ankle	4.0236	0.0022	0.9893	7.0578	True
Song_left_pocket	Song_right_hand	-1.5481	0.6684	-4.5824	1.4861	False
Song_left_pocket	Song_right_pocket	-0.4813	0.9	-3.5156	2.5529	False
Song_right_ankle	Song_right_hand	-5.5717	0.001	-8.6059	-2.5375	True
Song_right_ankle	Song_right_pocket	-4.5049	0.001	-7.5391	-1.4707	True
Song_right_hand	Song_right_pocket	1.0668	0.9	-1.9674	4.101	False

Figure 4 Post-Hoc Tukey Tables

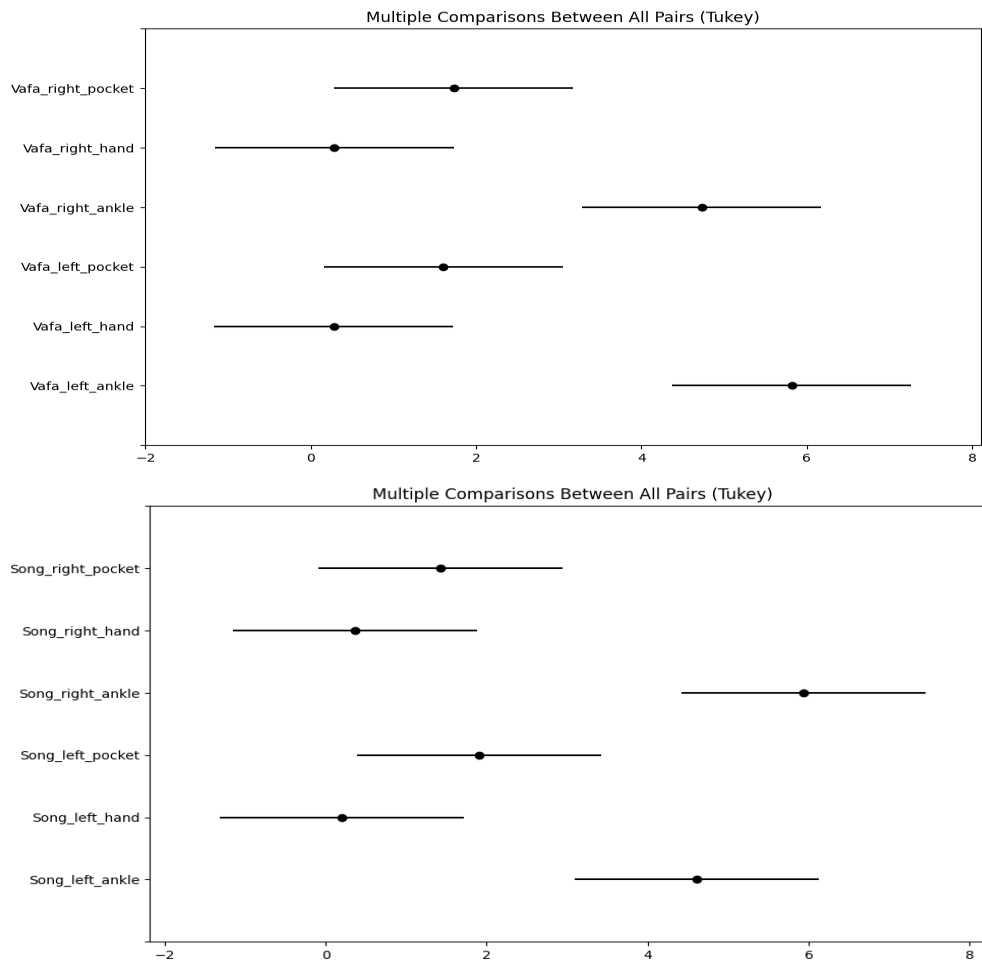


Figure 5 Post-Hoc Tukey Plots

According to Figure 4, the ANOVA test always fails to reject for different sides of the same placement, which means that there are no differences between the frequency of steps for the left and right foot. The low p-value from the ANOVA test was most likely the result of the difference between the placement of each foot. Hence, the conclusion that there are differences in the frequency of steps between two feet is rejected.

The analysis above also tells us that there is not always a difference in the mean of the frequency of steps between phone placements. On some occasions, the test indicates that there are differences between the placements, but sometimes it does not. We decided to perform the test for only the placements of the phone and disregard the aspects of which person or side it was recorded on. The result which is shown in Figure 6 indicates our conclusion from the ANOVA test still holds true to some extent. However, we must redefine the conclusion that there are differences between the frequency of steps recorded on the ankle and other areas. Performing the Post Hoc analysis on the other category yields the same result as the ANOVA test, which means there are differences in the frequency of steps between each person.

Multiple Comparison of Means - Tukey HSD, FWER=0.05						
group1	group2	meandiff	p-adj	lower	upper	reject
ankle	hand	-9.9832	0.001	-14.648	-5.3185	True
ankle	pocket	-7.2103	0.001	-11.8751	-2.5456	True
hand	pocket	2.7729	0.3445	-1.8919	7.4376	False

Figure 6 Post Hoc Analysis on Placement

Machine Learning:

As stated earlier, we want to see how well Machine Learning classifiers could classify which category the data goes to. Moreover, the result could be used as reinforcement for our conclusion drawn from the inferential statistics. We trained a model to classify two categories on two separate runs: one for classifying the placement of the phone, and one to classify which data belongs to each person.

Splitting training and validation data: The Y split of our data is the class, which includes the owner, location, and side. The rest of the summary data is the X. Due to the random nature of the *train_test_split()* function, we decided to compare a single split, with an average of 50 splits to get the most accurate results.

For each classifier, we had the data scaled using the *Standard Scaler* in the pipeline before giving it to the classifier.

Naive Bayes:

The result of the Bayesian Classifier is surprisingly good for classifying the placements of the phone. We were very surprised at how good the Bayesian Classifier performed. The average of 50 runs yielded relatively good results. Moving on to classifying a person, the classifier started to overfit, which resulted in a very bad result both after 1 run and after 50 runs:

K-Nearest Neighbors:

We ran kNN with 8 neighbours, this came from the process of trials and errors; of all the numbers we have tried, 8 seems to yield the best result. The result of kNN Classifier on classifying the placements of the phone is still relatively good compared to Naive Bayes:

However, for some strange reason, the validation score is higher than the training score. The only reasonable explanation that I could think of is due to chance and how the *train_test_split()* splits the data. The next reasonable step is to take the average of 50 runs which yields a more expected result. kNN performed very poorly for both after 1 run and after 50 runs when classifying a person.

Decision Tree:

For the decision tree, we ran the classifier with a max depth of 125. This number was chosen at random during trial and error, however, we found the result good enough for classifying the placements. The average of 50 runs also yields good results. The classifier did very poorly for classifying a person after both 1 run and 50 runs as it overfits the data just like Bayesian: We have tried different max depths, as the professor mentioned in the lecture that there is a point of diminishing return for the max depth value. However, we couldn't find any value that would do marginally better than 125.

Neural Networks:

For Neural Networks, most of the parameter was taken from the lecture slides, only the *max_iter* value was deliberately chosen as without it there would be an error. The result is very good for classifying the placement of the phone. The average of 50 runs also produces good results. Moving on to classifying person, the classifier did very poorly for both after 1 run and after 50 runs as it overfits the data just like Bayesian and Decision Tree.

Random Forest:

Compared to Neural Networks, Random Forest performed similarly in classifying the placement of the phone. As always, the Random Forest did very poorly on classifying the person:

Classifying Person	Naive Bayes	K-Nearest Neighbors	Decision Tree	Neural Networks	Random Forest
Training Score	0.925926	0.592593	1	0.925926	0.703704
Validation Score	0.22	0.1	0.33	0.22	0.33
Average Training Score (50 runs)	0.925926	0.518519	1	0.999259	0.55556
Average Validation Score (50 Runs)	0.555556	0.555556	0.3822	0.5733	0.33

Classifying Phone Placement	Naive Bayes	K-Nearest Neighbors	Decision Tree	Neural Networks	Random Forest
Training Score	1	0.962963	1	1	1
Validation Score	1	1	1	1	1
Average Training Score (50 runs)	1	1	1	1	1
Average Validation Score (50 Runs)	0.888889	0.888889	0.895556	0.9377778	0.88889

In general, all of the Machine Learning classifiers did a very good job of classifying the placement of the phone, which reinforced the conclusion that the placement of the phone does affect the frequency of steps. However, the same could not be said about classifying the person as all the models did very poorly. Part of the reason may be because we didn't have enough data.

Limitations

Our most significant limitation was the amount of data that we could realistically procure. Early on, we believed that we could get our friends to collect some data for us; however, that ended up falling apart. Another option that we discussed was using a website like Kaggle.com to find an accelerometer dataset. Had we done that; we could have more robust machine learning tools. Using an external dataset would come at a cost, as we do not know how they had obtained the data, it would skew our results. We could have used Kaggle data exclusively, and not our own, but this defeated the purpose of the project as we would not be gathering any data ourselves.

Another tough aspect is that we could not record both the sensor data from the linear accelerator and gyroscope in one go, this required us to either record the data from the other sensor and concatenate it to our current data or buy the pro version. Given the constraint of time, we don't think either option is worth it.

Conclusion

Through the above analysis of the data, we can conclude that the placement of the phone does affect how the sensors read the data. Although it should be noted that there are only differences between the frequency of steps generated from placing the sensor on the ankle and placing the sensor in the upper body. We could also conclude that there are differences between Song's step frequencies and Vafa's step frequencies, although that remains to be checked as more data is needed for us to confidently conclude that. In the end, I believe we did a pretty good job with the tools at our disposal and if only we could gather more data from either ourselves or other people, the results would have been much more concise. Moreover, there is a possibility of expanding the accuracy, if the app we used allowed us to record data from two sensors at once or we could record more data in the future and concatenate it to our current data.

Project Experience Summary:

Song Tung Nguyen:

- Analyzed sensor data using inferential statistical tools to draw conclusions from the data.
- Manipulated Machine Learning Model to make sense of the data and draw conclusions from it.
- Collaborated with another member to design and carry out the analysis of walking data.

Vafa Dehghan Saei:

- Worked in a group of 2 to collect and analyze walking data using different machine learning techniques.
- Researched existing work on accelerometer data analytics to get a better understanding of how to filter noise out of accelerometer data.
- Collected data in a controlled environment to minimize the effects of outside forces.
- Performed the entire ETL process on a set of accelerometer data to convert the data to a more format that is easier to work with.

References

Goto, K., Kawabat, M., Fukusaki, C., Sasaki, K., Mizushina, T., & Ishii, N. (2016). Three-dimensional Acceleration During Running In Trained Athletes And Untrained Subjects. 34 *International Conference of Biomechanics in Sport*. Tsukuba: International Society of Biomechanics in Sports.

Fortune, E., Lugade, V., Morrow, M., & Kaufman, K. (2014). Validity of using tri-axial accelerometers to measure human movement - Part II: Step counts at a wide range of gait velocities. *Medical engineering & physics*, 36(6), 659–669.
<https://doi.org/10.1016/j.medengphy.2014.02.006>

Yousefian, Maria et al. "FCEstimator : Self-Monitoring Foot Clearance App to Assess Risk of Falls Using a Smartphone." (2018).