ML4QS: Assignment 1

Martin Banchero and Dennis Dekker

VU, Amsterdam, NL

1

2 Chapter 2

2.1 Pen and Paper 1

Differences between sensory data across multiple users can be the result of:

1. Difference in used devices.
2. Different environments of the users.
3. Different ways users interact with the devices.
4. Different activities users do.

2.2 Pen and Paper 2

Four criteria that play a role in deciding on the granularity of the measurement of the dataset:

1. Information loss when using higher granularity.
2. High granularity results in small amount of instances.
3. Lower granularity leads to more variance.
4. More outliers in lower granularity.

2.3 Pen and Paper 3

Next to the two set tasks of the assignment, we could think of different other tasks that can be performed using the crowdsignals data:

- Use **unsupervised learning** to look for structure in the data, This can be achieved by performing clustering, this can help to get an insight of the different attributes. For example finding groups of user that have the same patterns in doing certain activities, like working out in the evening. This information can be used to make suggestions related to working out at these timepoints.
- Use **reinforcement learning**. The goal of this task is to find which actions lead to a better reward. For example, learn what training scheme give the best improvements for an user.

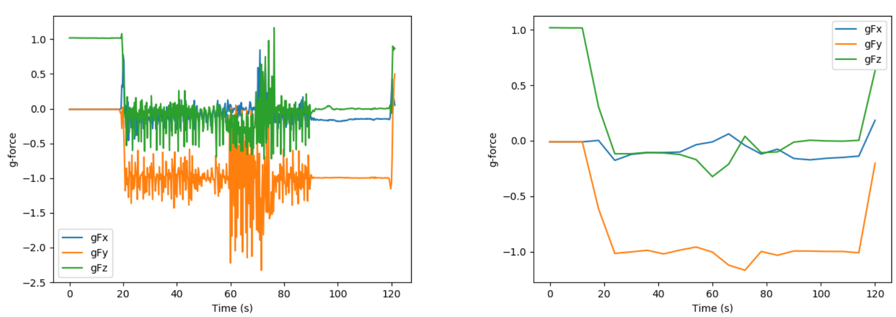
2.4 Coding 1

Using the app Physics Toolbox Sensor Suite, we created a dataset with measurements form different sensors. In total, we used 7 sensors: g-Force (3), linear accelerometer (3), gyroscope (3), barometer, magnetometer (3), inclinometer (3) and a sound intensity meter. This resulted in 17 measurements per timepoint, as some sensor had multiple outputs (amount indicated in brackets if more than 1). The frequency of data collection was set to as fast as the device allowed, what resulted in multiple measurements per 10 ms. However, the frequency was not constant, so some intervals had more measurements than others. Still, there was at least 1 measurement per 10 ms.

Plots and data description The data is displayed in figures 1 to 7, the activities and associated intervals in table 1.

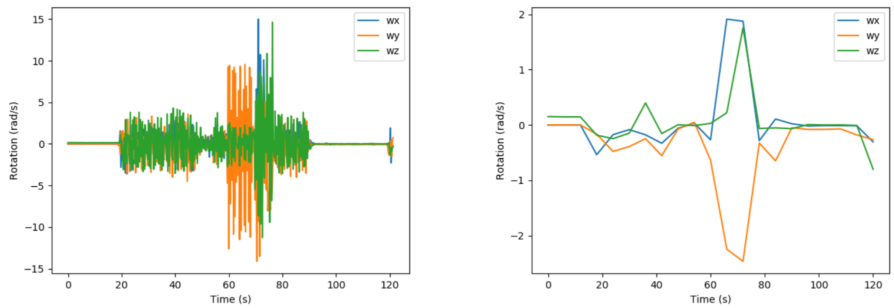
Table 1. Activities per interval for generated data. Timepoints in seconds.

Start	End	Activity
0	20	Baseline
20	60	Walk
60	70	Jump
70	75	Run
75	90	Walk
90	120	Stand



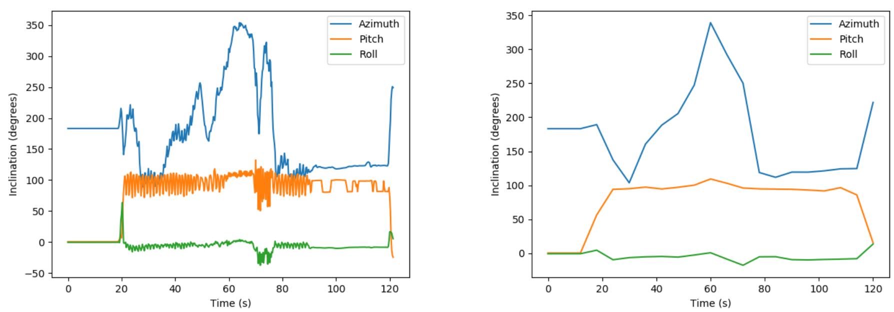
(a) 250 ms (b) 6000 ms

Fig. 1. G-force measurements



(a) 250 ms (b) 6000 ms

Fig. 2. Gyrometer measurements



(a) 250 ms (b) 6000 ms

Fig. 3. Inclinometer measurements

3 Chapter 3

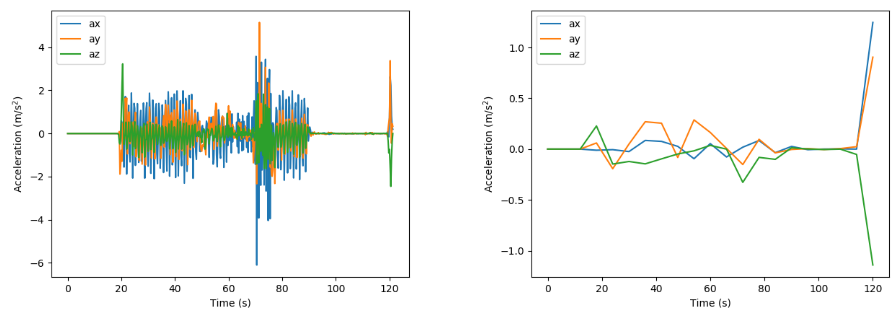
3.1 Pen and Paper 2

We have seen two types of outlier detection algorithms: distance and distribution based. In what situations would it be better to apply a distance based outlier detection algorithm over a distribution-based approach?

For the distribution-based algorithm the distribution of the data has to be known and is applied to individual attributes.

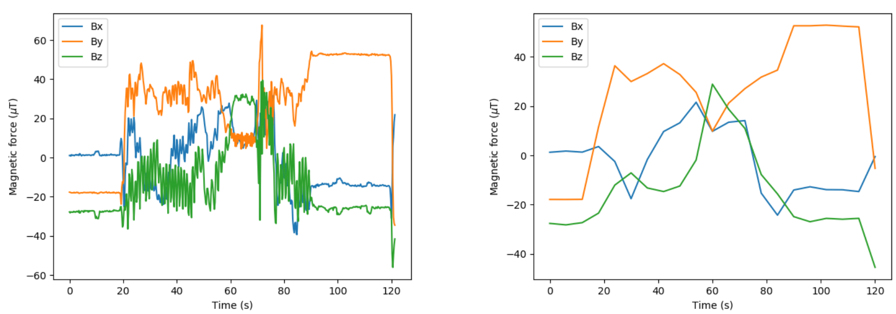
3.2 Pen and Paper 4

The local outlier factor (LOF) algorithm is quite complex. Find out what the computational complexity of the algorithm is and discuss ways to improve the scalability of the approach.



(a) 250 ms (b) 6000 ms

Fig. 4. Linear accelerometer measurements



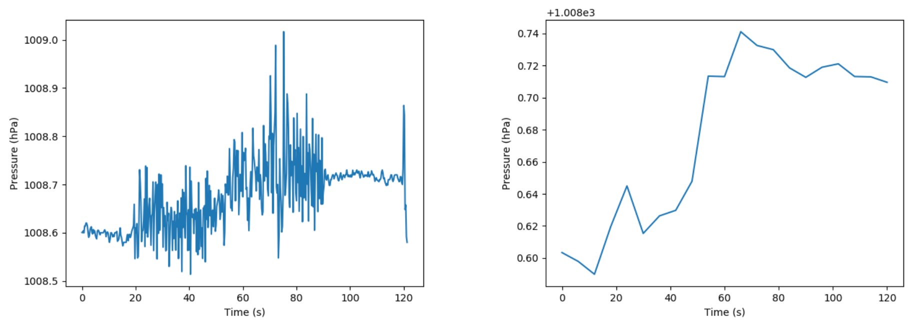
(a) 250 ms (b) 6000 ms

Fig. 5. Magnetometer measurements

The order of complexity of the algorithm is $O(n^2)$ where n is the size of the dataset. The algorithm is computational intensive because compute the LOF for every object in the dataset. Different approaches can be used to improved the scalability [1]. One way to improve the algorithm can be reducing the amount of distances to be compute [2]. Other approach using computational power consist in GPU platform to accelerate the LOF [1]

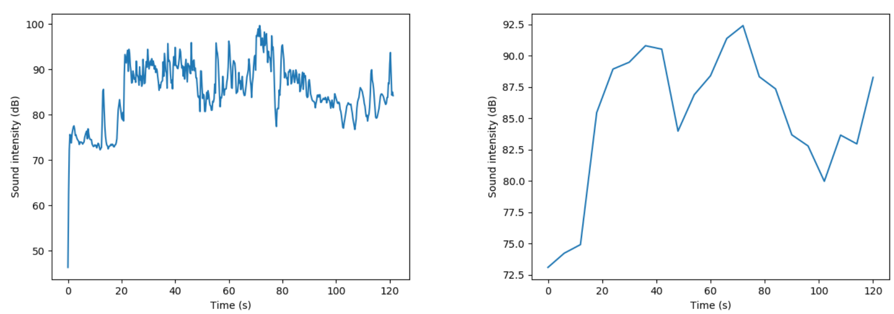
3.3 Coding 3

Use a model-based approach to impute the heart rate. The crowdsignal data had a lot of missing values for the heartrate, see Table 2. We applied the Kalman filter in order to predict the missing values, of which the first rows are displayed in the table.



(a) 250 ms (b) 6000 ms

Fig. 6. Barometer measurements



(a) 250 ms (b) 6000 ms

Fig. 7. Sound intensity measurements

3.4 Coding 4

Similarly to what we have done for our crowdsignals dataset, apply the techniques that have been discussed in this chapter to the dataset you have collected yourself. Write down your observations and argue for certain choices you have made.

In order to detect outliers in our data, we applied the four different outlier detection methods discussed in this chapter: Chauvenet, Mixture model, simple distance based and local outlier factor. In order to see the difference in performance, we applied these methods on the atmospheric pressure column. The results are displayed in Figure

It can clearly be seen that the different methods give more or less the same results. Also, if you look at the values for the pressure, most of the extreme values are marked as an outlier, which makes perfect sense.

Timepoint	p	Chauvenet	Mixture_model	simple_dist_outlier	lof
69.5	1008.676479	FALSE	FALSE	FALSE	1.005072586
69.75	1008.8	FALSE	FALSE	FALSE	1.001477085
70	1008.793784	FALSE	FALSE	FALSE	1.004384947
70.25	1008.925278	TRUE	FALSE	FALSE	1.094158569
70.5	1008.844795	FALSE	FALSE	FALSE	1.043018522
70.75	1008.682778	FALSE	FALSE	FALSE	0.998209114
71	1008.806056	FALSE	FALSE	FALSE	1.013601515
71.25	1008.64038	FALSE	FALSE	FALSE	0.993864357
71.5	1008.78	FALSE	FALSE	FALSE	1.00688412
71.75	1008.8125	FALSE	FALSE	FALSE	1.015112536
72	1008.850556	FALSE	FALSE	FALSE	1.040175586
72.25	1008.988493	TRUE	TRUE	TRUE	1.123811496
72.5	1008.695	FALSE	FALSE	FALSE	0.99793226
72.75	1008.682105	FALSE	FALSE	FALSE	1.001541195
73	1008.7	FALSE	FALSE	FALSE	0.998229527
73.25	1008.547778	FALSE	FALSE	FALSE	1.039044587
73.5	1008.591918	FALSE	FALSE	FALSE	0.987895258
73.75	1008.753333	FALSE	FALSE	FALSE	1.008320946
74	1008.692329	FALSE	FALSE	FALSE	1.001907205
74.25	1008.664533	FALSE	FALSE	FALSE	0.996596815
74.5	1008.6016	FALSE	FALSE	FALSE	0.990939011
74.75	1008.612778	FALSE	FALSE	FALSE	0.934598129
75	1008.674932	FALSE	FALSE	FALSE	1.003496353
75.25	1009.016667	TRUE	TRUE	TRUE	1.111257019
75.5	1008.778767	FALSE	FALSE	FALSE	1.00769476
75.75	1008.677532	FALSE	FALSE	FALSE	1.004014243
76	1008.692	FALSE	FALSE	FALSE	1.002347234

Fig. 8. Subset of the results of the outlier detection methods. Outliers of the Chauvenet, Mixture model and simple distance methods on the pressure data p are indicated with *TRUE*. The value for the local outlier factor is displayed in the column *lof*. The higher the score, the more likely it is that the point is an outlier.

Table 2. Show the first 10 rows and three columns of the dataset *chapter2_result.csv* after Kalman filter.

Point	date-time	Heart rate	Kalman prediction
0	2016-02-08 18:28:25.656222395	NaN	73.802813
1	2016-02-08 18:28:25.906222395	NaN	73.802813
2	2016-02-08 18:28:26.156222395	NaN	73.802813
3	2016-02-08 18:28:26.406222395	NaN	73.802813
4	2016-02-08 18:28:26.656222395	NaN	73.802813
5	2016-02-08 18:28:26.906222395	159.5	154.567223
6	2016-02-08 18:28:27.156222395	NaN	154.567223
7	2016-02-08 18:28:27.406222395	158.0	157.564539
8	2016-02-08 18:28:27.656222395	156.0	156.323020
9	2016-02-08 18:28:27.906222395	154.0	154.487634

4 Chapter 4

4.1 Pen and Paper 1

We have seen several functions that summarize numerical values within the time domain to a single number (i.e. mean, standard deviation, minimum, and maximum). Provide an example for all four functions that shows where that specific form of summarization can be useful.

1. **Mean:** To get an estimate of the actual values over the window, you can use the mean to compensate for measurement errors.
2. **Standard deviation:** To get an estimate of the variability of the data values in a specific window.
3. **Minimum:** If you want to get an indication of the resting blood pressure of a person. Can be used to see if a person suffers from high blood pressure, as the resting blood pressure is a good indication of this.
4. **Maximum:** If you want to get an indication of the maximum heart rate of a person, for example

4.2 Pen and Paper 6

Besides generic features, we might also have dedicated features we engineer for a specific domain. Imagine that we want to learn a model that predicts someones mood based on the amount of social activity. Define three dedicated features that can be useful in this context based on measurements we can potentially be collected from the mobile phone.

1. Amount of time on social media apps.
2. Amount of messages send through messenger apps (like Whatsapp, Facebook, etc.).
3. Amount of calls made .

4.3 Pen and Paper 7

We have discussed dedicated approaches for handling text based data. One aspect we discussed was to perform stemming on the words to make sure all conjugates of verbs or plural forms of nouns are considered as the same word. Think of one advantage and one disadvantage of using stemming.

Advantage: It is easier to make predictions based on features, as there are less features and there is more data per feature.

Disadvantage: The information in conjugates or plural form of words is lost. Also the sense of time is lost as all verb are in the present form.

4.4 Coding 1

Explore the frequency domain features for the crowdsignals dataset in more detail, consider the individual frequencies for the different measurements and see whether you can find interesting patterns. Do you see consistent amplitudes of certain frequencies during the same activities? And how do the amplitudes differ for the different activities?

Unfortunately, the coding of this part didn't work out. We have to investigate why this part is not working, or rewrite the script from scratch. We hope to see specific frequencies for each activitiy, especially for running and walking. These frequencies would be a good feature to use for predicting these activities in the end.

4.5 Coding 2

Implement at least two additional metrics in the time domain and the frequency domain in addition to the ones already present in the data (e.g. the ones you have identified in a previous question). Calculate them for the crowdsignals data and discuss their usefulness.

We couldn't implement this part yet, see subsection 4.4.

References

[1] Malak Alshawabkeh, Byunghyun Jang, and David Kaeli. *Accelerating the Local Outlier Factor Algorithm on a GPU for Intrusion Detection Systems.* 2010. ISBN: 9781605589350. URL: `http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.410.2365&rep=rep1&type=pdf`.

[2] Sunil Arya et al. *An Optimal Algorithm for Approximate Nearest Neighbor Searching in Fixed Dimensions.* Tech. rep. 6. 1994, pp. 891–923. URL: `http://delivery.acm.org.vu-nl.idm.oclc.org/10.1145/300000/293348/p891-arya.pdf?ip=154.59.124.111&id=293348&acc=ACTIVE%20SERVICE&key=0C390721DC3021FF.5F9071D3233F7DA5.4D4702B0C3E38B35.4D4702B0C3E38B35&__acm__=1560112313_8e845b227d1b25a979f53fdb6af54473`.