ML4QS - Assignment 1

Tommie Kerstens and Tamvakis Charalampos

VU University Amsterdam, De Boelelaan 1105, 1081 HV Amsterdam, the Netherlands

Abstract. This document is report produced by group 11 for the first assignment in the VU University Course - Machine Learning for the Quantified Self for the MSc AI programme. In this paper we apply data analysis and machine learning techniques to extract information from sensory data which was both provided and collected by us. We also perform modeling to impute missing values and perform feature engineering to further improve the data for further analysis.

1 Theoretical Assignments

1.1 Pen and Paper for Chapter 2

Q1. Three potential causes for differences between sensory values obtained are:

1. Differences caused by manufacturing processes and calibration.
2. Differences in in the environment the sensors operate in.
3. Desired differences measured due to the way the subject uses the device in which the sensors are embedded.

Q2. Four criteria that play a role in deciding on the granularity for the measurements of a dataset are:

1. **The task:** Some tasks might need more granularity than others, as we saw in chapter 1 in the example of classifying movement as "Walking". A delta t of 1 minute did not allow the data to encode the needed information for this task.
2. **The noise level:** Lower granularities (higher values for delta t) cause a smoothing effect on the data. This effect might be desired if the data is noisy, but could also smooth out important details in the data.
3. **Available memory:** Higher levels of granularity cause larger datasets that require more memory.
4. **Cost of storage:** In the case of very large datasets that also might need to be available online, one needs to consider the cost of storing the datasets.
5. **Available computational resources**: Very granular and detailed data might cause a model to be too large or complex to compute requiring faster hardware which might or might not be available to the researcher.

Q3. Other possible machine learning tasks that could be performed on the Crowdsignals dataset and why they are relevant to the support of a user.

1. **Clustering Task:** Potentially discovering patterns (clusters) in the data that might point at specific parameters (e.g.: locations, movements, heart rate zones) that bring about some (un-)desired consequence.
2. **Outlier Detection:** Being able to more reliably filter out outliers during the preprocessing of the data could improve the accuracy of later learned models.

1.2 Pen and Paper for Chapter 3

Q2. In the case we know the phenomena we are observing does not represent a normal distribution we should use distance-based outlier detection.

Q4. The LOF algorithm is an algorithm that performs well in lower dimensional data. When increasing dimensionality it can become highly exponential in terms of complexity due to it's nearest-neighbor approaches for finding outliers.Since it uses minimum distances between neighbors to detect if all corresponding points are outliers, this can result up to a complexity of $\mathbf{O}(n^2)$ since we compare a set of nearest neighbors for each data point, in a naive approach. We can reduce that to **O(nlogn)** complexity by introducing R-trees and spatial tree indexing [2]. Another approach would be to use ensembles [3]. One final approach that we thought of is also to use simple clustering methods such as k-mean, and detect outliers for specific clusters, although this approach would show improvement in low dimensional data.

1.3 Pen and Paper for Chapter 4

Q1. The *minimum* and *maximum* can be useful for the normalization of the data. Also in cases where values are supposed to stay within certain bounds (e.g.: a centrifuge which is not allowed to exceed a certain maximum or drop below a certain minimum), the summarization of a minimum or maximum can be used. The *mean* and *standard* deviation can be useful for the imputation of missing values.

Q3. One approach would be to aggregate more granularly to reduce the amount of data which needs to be processed. An extension to this approach would be to iteratively increase the granularity on sections that contain certain temporal target patterns, to identify at which granularity these patterns still exist and better identify predictors of these target patters.

Q4.
1. *Bandwidth:* is the difference between the upper and lower frequencies in a continuous set of frequencies. In communications the bandwidth of a signal is a determinant its capacity for communication.
2. *Lowest Frequency Amplitude:* Music producers and other audio experts need a flat spectrum to optimally listen to sound with the least amount of distortion.

Q6.
1. Amount of phone usage. Some users might become very unsocial and spend larger amounts of time on their mobile phones not making any calls.
2. Quantity of spectral activity within the range of human voices. When this activity is lower then usual the user might be isolated, or this might signal some other mood.
3. The use of some sort of Natural Language Processing (NLP) to extract data about the mood of the user, for example by making use of some type of topic modeling like Latent Dirichlet Allocation (LDA) [1].

Q7. One advantage of stemming is that it reduces the complexity of the NLP task, as you end up with less words. One disadvantage of this approach is that this process may cause the loss of detail in the meaning of the sentence.

Q8. Because of the imbalance in the representation of the topics the model is tempted to become insensitive to the underrepresented (2%) topic. We could alleviate this problem by introducing more instances of the underrepresented topic in our training data, or by punishing the model harder for misclassification of the underrepresented topic.

1.4 Coding for Chapter 2

Task 1 We used the smartphone application SensorLog to gather data. SensorLog was installed on two separate devices that we used to collect data. The following sensors were used to gather the data from: GPS - Accelerometer - Gyroscope - Proximity Sensor - Light Sensor - Step Detector - Linear Acceleration - Rotation Vector. The data was first transformed from a log file into multiple **csv** files where the headers were of the format:

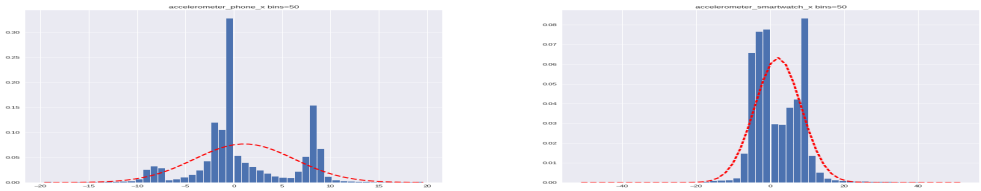
label|**value vector**|**timestamp**.

We then extracted these labels according to their timestamp into a different file and created the dataset for different time frames e.g. `100,250,1000 ms`. The choice was based that we need a small and medium granularity, based on the start and end of the task labeled. A notable observation is that our measurements have a periodic pattern, as also as that the rotation is reversed for our 2 tasks. After preparing the data we utilized the code that was provided by the course to plot the data. The figures can be seen in the plots below.

Task 2 Comparing with the crowdsignals dataset, 2 notable labels that can be discussed here is the walking/running. For these 2 labels we see similar patterns in the accelerometer and different pattern for the gyroscope, since we were walking along all axis (x,y,z where they correspond to blue,orange,green colors and the rotation has also theta,phi angles over the 2 axis with red purple colors).

1.5 Coding for Chapter 3

Task 1 Histograms (a) and (b) displayed in figure 1 show the probability distribution of samples in the raw data. The red line indicates a normal distribution based on the same data displayed. The images indicate that the raw data is not normal distributed, and applying an outlier detection filter based on the Chauvenets Criterion could possible filter out more from the data then only outliers. Images c and d in the same figure show this same type of analysis, but now performed on the data aggregated by means of averaging over 1000ms. After aggregating we still do not recognize the data as being normally distributed. We do however see that the process of aggregation by averaging reduced the standard deviation from 5.18 to 4.51 and from 6.30 to 5.36 for the phone and smartwatch respectively, this however is easy to declare given the averaging of values in the process.

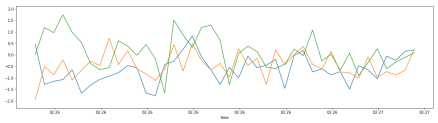


(a) Phone Accelerometer (b) Smartwatch Accelerometer

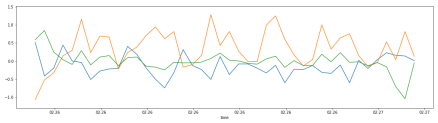
Fig. 1: Raw data histograms for the accelerometers of phone and watch over the x axis. The red line is the ideal normal distribution derived from the measurement's mean and standard deviation.

(a) Phone Accelerometer (b) Smartwatch Accelerometer

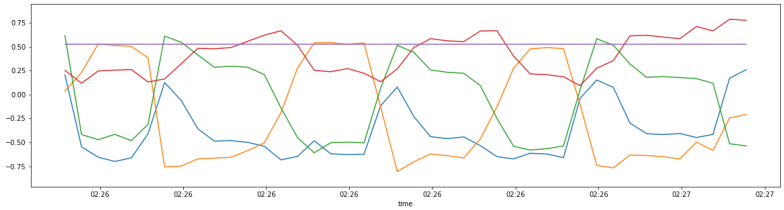
Fig. 2: Aggregated histograms for the above measurements. We can still see that these measurements do not follow a normal distribution due to the large count of outliers and the granularity of 1 second.



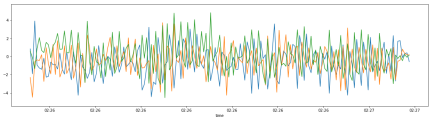
(a) 1000ms Accelerometer



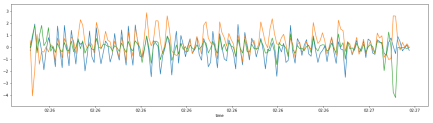
(b) 1000ms Gyroscope



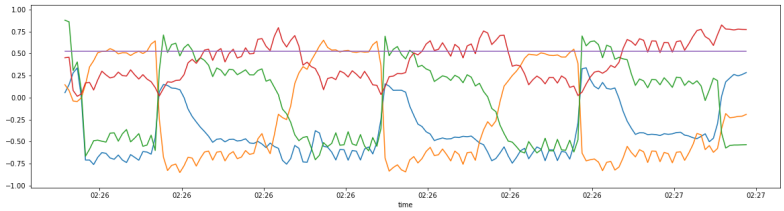
(c) 1000ms Rotation



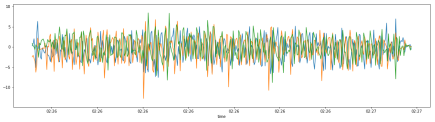
(d) 250ms Accelerometer



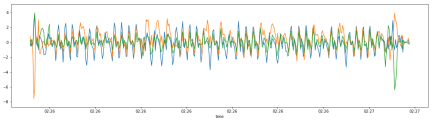
(e) 250ms Gyroscope



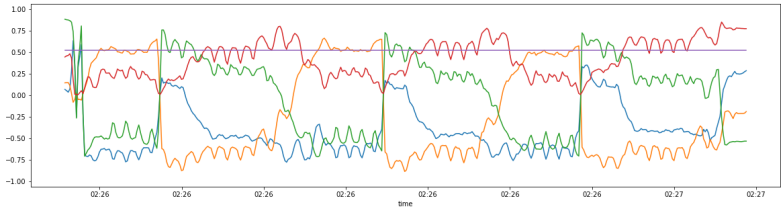
(f) 250ms Rotation



(g) 100ms Accelerometer

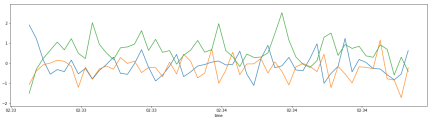


(h) 100ms Gyroscope

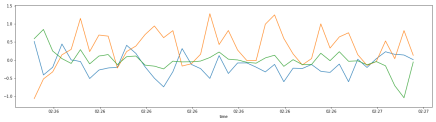


(i) 100ms Rotation

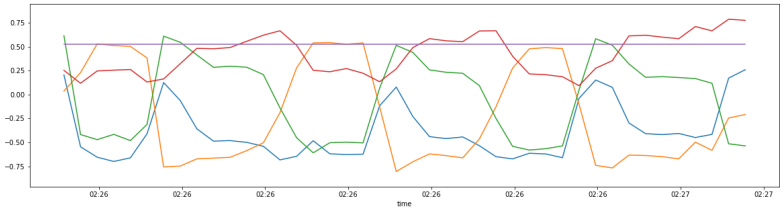
Fig. 3: Data obtained by the accelerometer, gyroscope and rotation while walking down the stairs aggregated over 1000, 250 and 100 milliseconds.



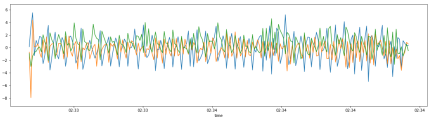
(a) 1000ms Accelerometer



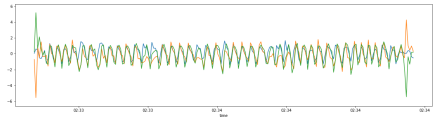
(b) 1000ms Gyroscope



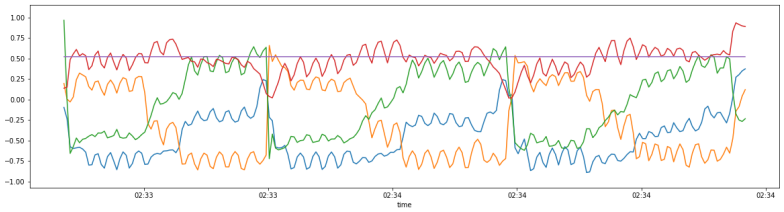
(c) 1000ms Rotation



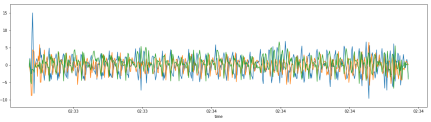
(d) 250ms Accelerometer



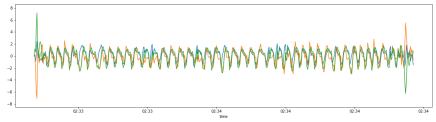
(e) 250ms Gyroscope



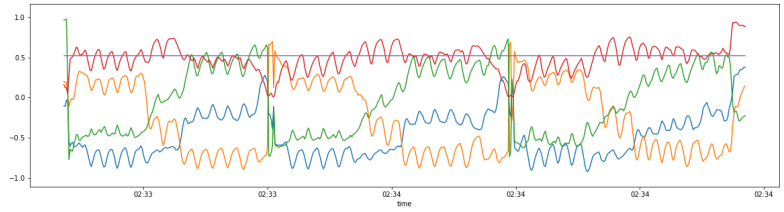
(f) 250ms Rotation



(g) 100ms Accelerometer



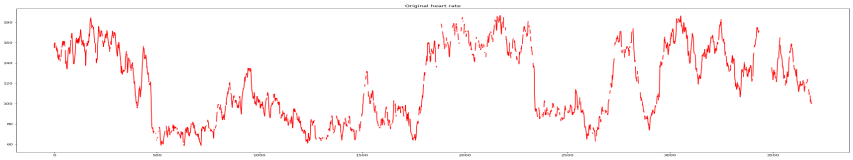
(h) 100ms Gyroscope



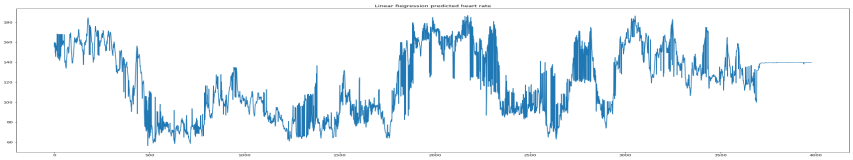
(i) 100ms Rotation

Fig. 4: Data obtained by the accelerometer, gyroscope and rotation while walking up the stairs aggregated over 1000, 250 and 100 milliseconds.

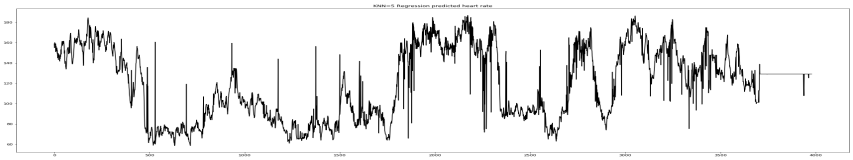
Task 2 For this task, we implemented 2 model based approaches to impute the missing values of the heart rate from the crowdsignal dataset. We used 3 models:



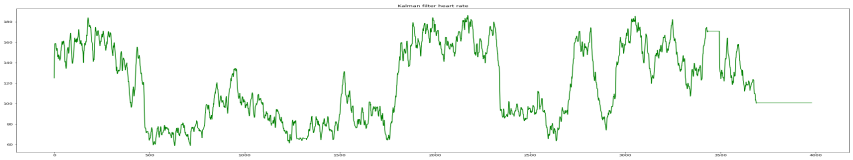
(a) Original Data



(b) Linear Regression



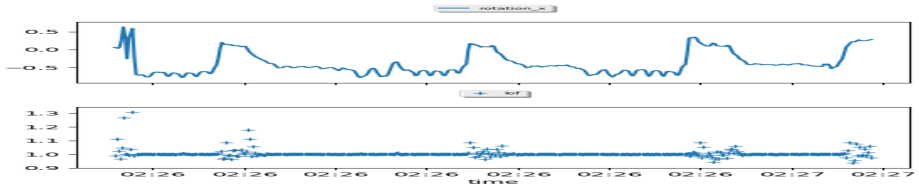
(c) Nearest Neighbor Regression



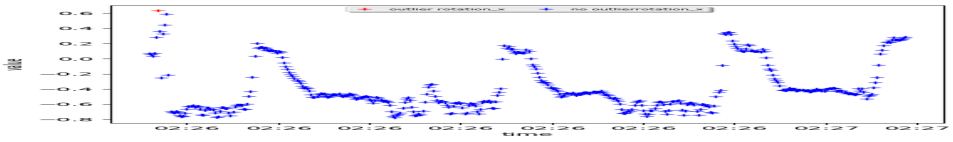
(d) Kalman Filter

Our first model works pointwise therefore in the missing value areas showed a lot of different imputations, while the knn was close to the kalman filter. The kalman filter has a more smooth approximation on how the missing data would be approximated in comparison to the surrounding values.

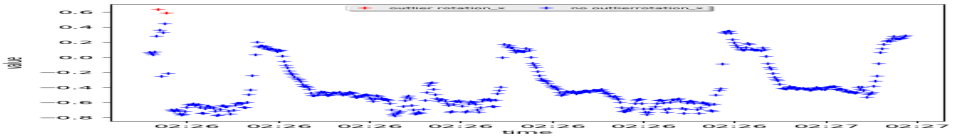
Task 3 We performed similar techniques described in this chapter for our sensory data. Since we don't have any missing values in our data we discarded some for the rotation theta and try to predict on those. We used outlier detection and filtering according to the figures below for the rotation_theta measure:



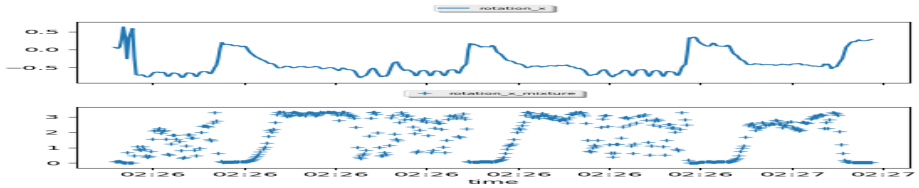
(a) Local outlier Factor



(b) Distribution outlier Detection

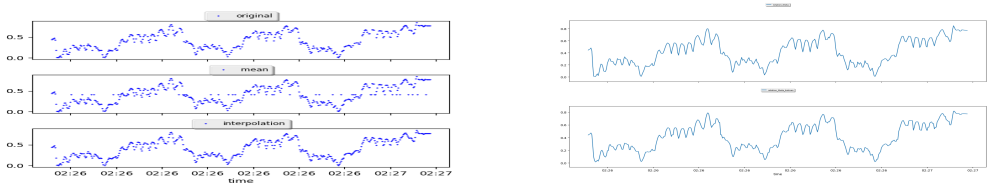


(c) Distance-Based Outlier Detection



(d) Distance-Based Outlier Detection

Fig. 5: Several applied outlier detection techniques on the rotation over the theta axis of the phone. Only the distance-based method showed some outliers.



(a) Simple filtering

(b) Kalman Filter

Fig. 6: It can be shown that the kalman filter smoothed some of the values in comparison to the other filters.

1.6 Coding for Chapter 4

Task 1 As we are out of time and space to finish this assignment we will include the produced graphs for this last coding task in figure 7.

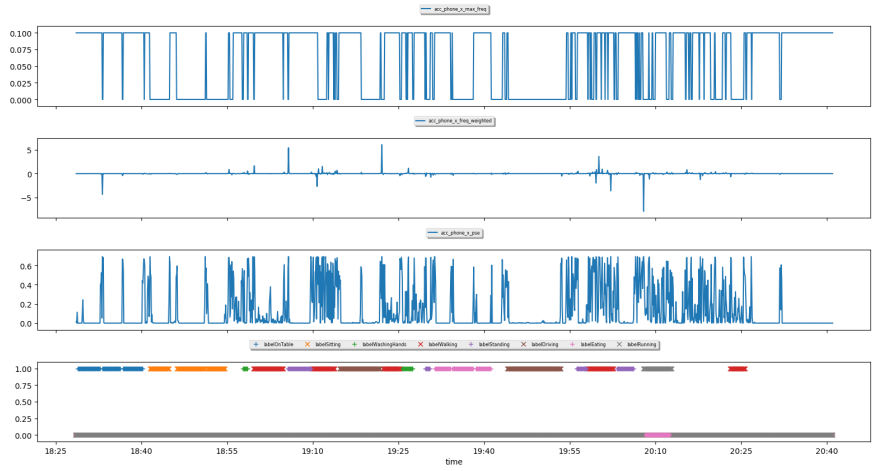


Fig. 7: Fourier analysis. Frequency distribution over time with a window of 5000 milliseconds.

References

1. David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.
2. Ke Zhang, Marcus Hutter, and Huidong Jin. A new local distance-based outlier detection approach for scattered real-world data. *Advances in knowledge discovery and data mining*, pages 813–822, 2009.
3. Arthur Zimek, Ricardo J.G.B. Campello, and Jörg Sander. Ensembles for unsupervised outlier detection: Challenges and research questions a position paper. *SIGKDD Explor. Newsl.*, 15(1):11–22, March 2014.