MCM ML 520 Assignment 3 - Gesture Recognition

Gesture Recognition

Dataset

- The dataset contains different arm gestures from different participants.
  - Recording device and wearing position: smartwatch (LG Watch G) running Android Wear, worn at right wrist.
  - Recorded data: 3 axes acceleration measured in G [9.81m/s^2], with a target sampling rate of 50Hz.
  - Gestures, participants and samples: 8 gestures, 9 participants. Each participant performed each gesture 30 times, which results in a total of 240 samples per participants, 270 samples per gesture and 2160 samples in total in the data set.
  - The publication describing the dataset in more detail can be found here: https://ambientintelligence.aalto.fi/team_old/findling/pdfs/publications/Kefer_16_ComparingPlacementTwo.pdf
- Provided csv files:
  - Acceleration recordings are split per axis into separate csv files (one file per axis). This implies that e.g. the Nth line of each file belong to the same gesture, participant and sample and represent the according x, y and z axis acceleration recordings.
  - Csv file columns are in this order:
    1. gesture
    2. participantNr
    3. sampleNr
    4. N acceleration values (different lengths per sample)

Gestures in the dataset

Task

- Goal is to build (a) successful gesture recognition model(s). The model should learn to distinguish between N gesture acceleration recording, then be able to decide for new recordings which gesture it shows. Load data,

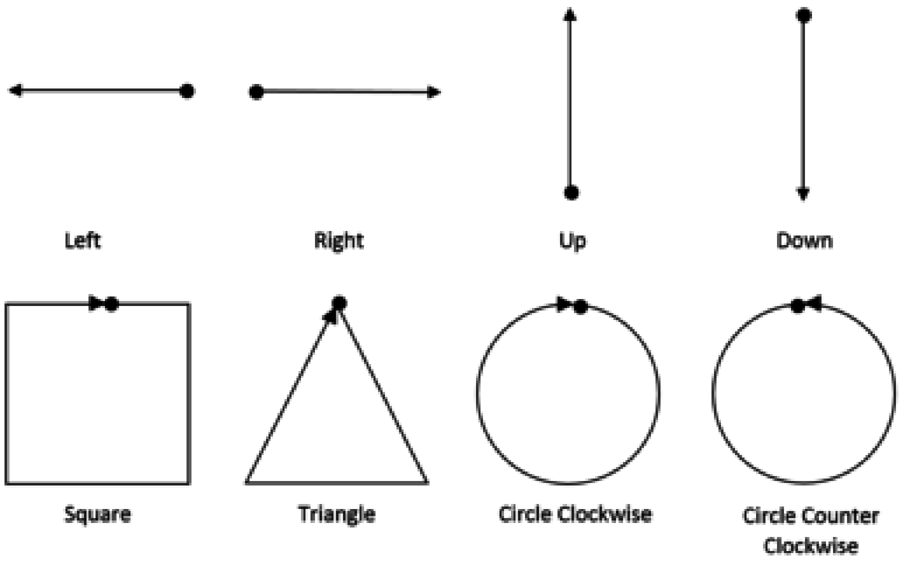


Figure 1: Gestures in the dataset.

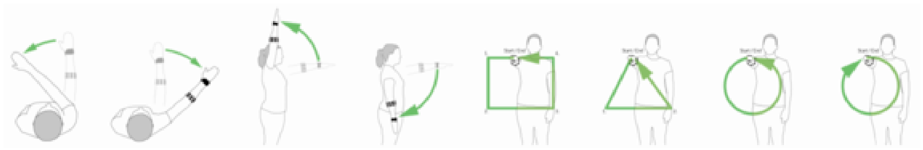


Figure 2: Gestures in the dataset.

preprocess it, and extract features (see hints below).

  - Do a gallery dependent data partitioning and train models using cross validation. Compare different models and feature extraction approaches by their cross validation results, and for the selected “best” model show at least the confusion matrix for held-back test partition over different gestures. Discuss what the best feature extraction/model configuration is you found, and if there are gestures that are harder to distinguish than others.
  - **Bonus objective**: do gallery independent data partitioning instead using leave subject out cross validation (LSOCV, see slides).

Hints

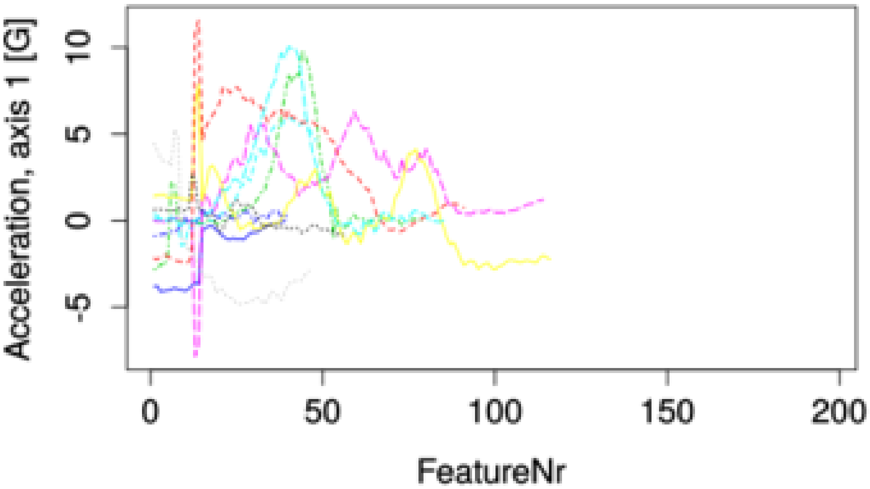
- Use vectorized commands and clever parametrization of read.table to load all csv files at once.
- Recorded acceleration should probably be filtered to reduce noise. The rolling function of pandas can help here. Consider e.g. running mean, running median, or SV filters. Hint: you probably need to apply a stronger filter than would seem intuitive at the beginning. Think about *which info* needs to remain in the task - it might be less that you would intuitively guess. Check that you still have the required information in the data after filtering (e.g. using plots). Again, try to use vectorized commands.
- If you plan on using acceleration values themselves as features, different lengths of recordings are going to be problematic. You can interpolate series to uniform length . Again, try to use vectorized commands. Be aware that with interpolation to a uniform amount of features you also lose the implicit information of how long the recording was (which was expressed by the amount of samples before). The length of the original recording can therefore become a numeric feature itself. Hint: you probably need less interpolated acceleration values than you would expect. Usually, human body motion can produce up to a max of about 20Hz of movement frequencies, so using e.g. 100Hz recordings does not necessarily add any information you can use. And the important acceleration information to distinguish gestures is likely well below 20Hz, so a couple of values per gestures probably already do the job.
- Acceleration values of all axes can be used directly as one sample (concatenate all axis into one row).
- Feature extraction is not limited to those discussed in the lecture. But for a start:
  - Frequency power and phase
  - Autocorrelation of acceleration values
  - Wavelet components
  - All of the above can be applied on the integrations of acceleration as well (position <–> speed <–> acceleration).

Hand-in

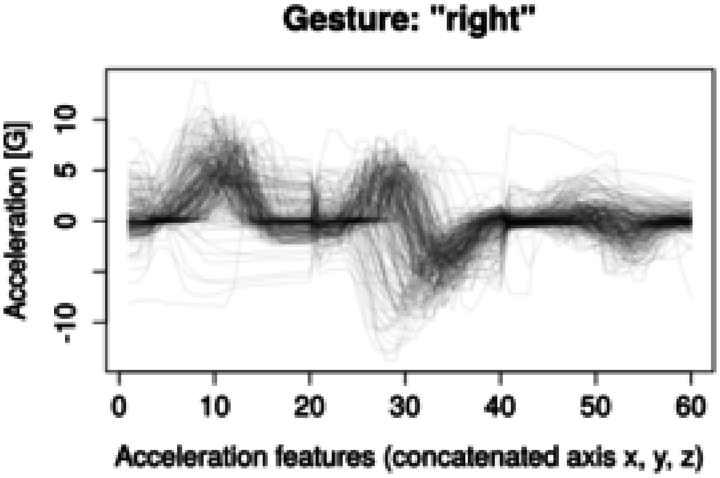
- You can chose tools as you like and can span your processing chain across multiple platforms/languages. Hand in your code and the report describing what you did and found.
- In the report: document everything so that **your system could be rebuilt from using the report only**. The report is highly important for this assignment: everything you want reviewers to *understand* needs to be stated in it:
  - Flowchart/block diagram of the processing chain you use, containing each step from the original data to model training and evaluation performance testing.
  - State clearly how you do data partitioning.
  - State clearly which feature extraction/transformations with which parameters you apply.
  - State and compare performance of different models/approaches.
  - State which model you would chose for a real world application, why you would do so, and its performance (confusion matrix, ...).
  - Don't forget to state your thoughts/findings on your data at each point in the processing chain (why you apply transformations, what you see, etc).

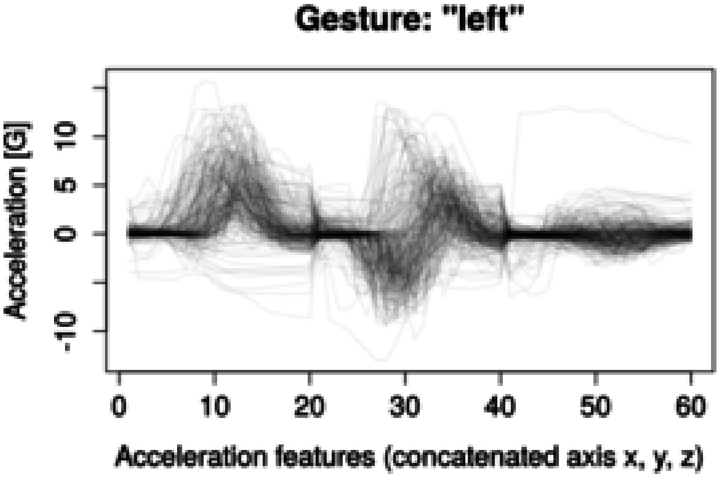
Examples

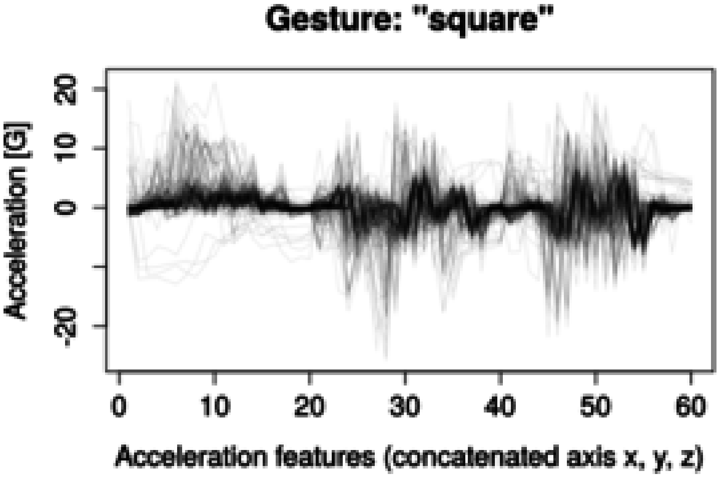
Visualizing your data/results in a similar way will be beneficial. Gesture samples:



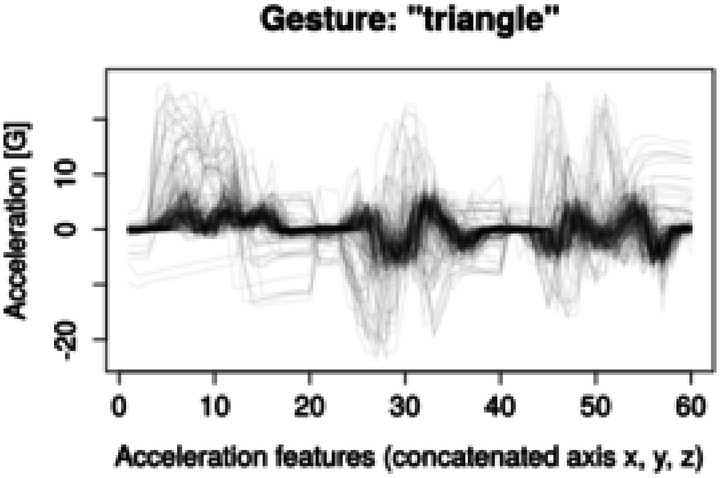
Filtered and interpolated gesture samples:





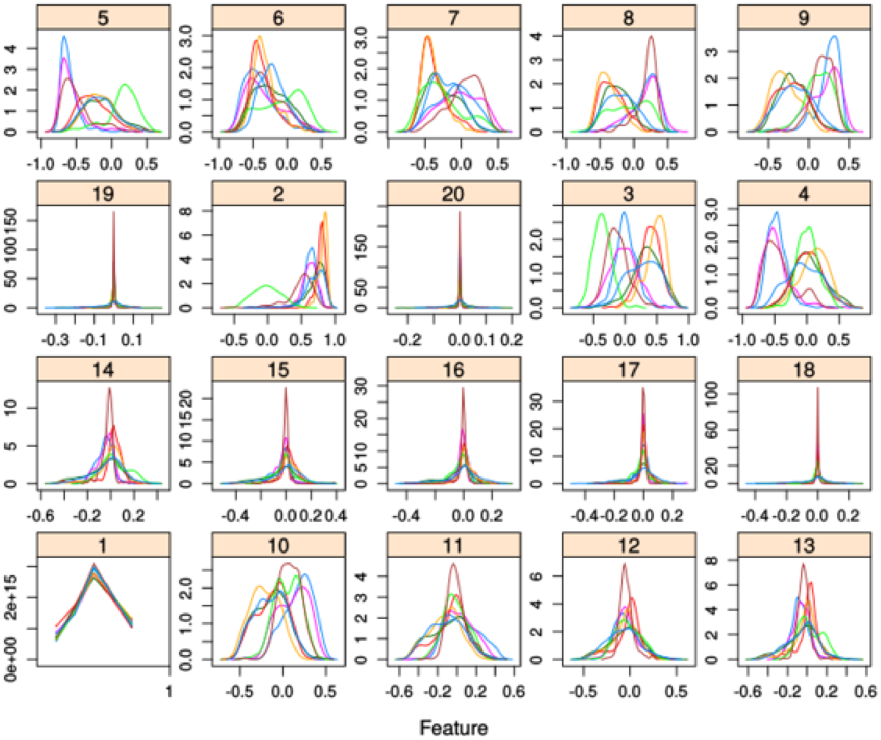


Gesture: "square"

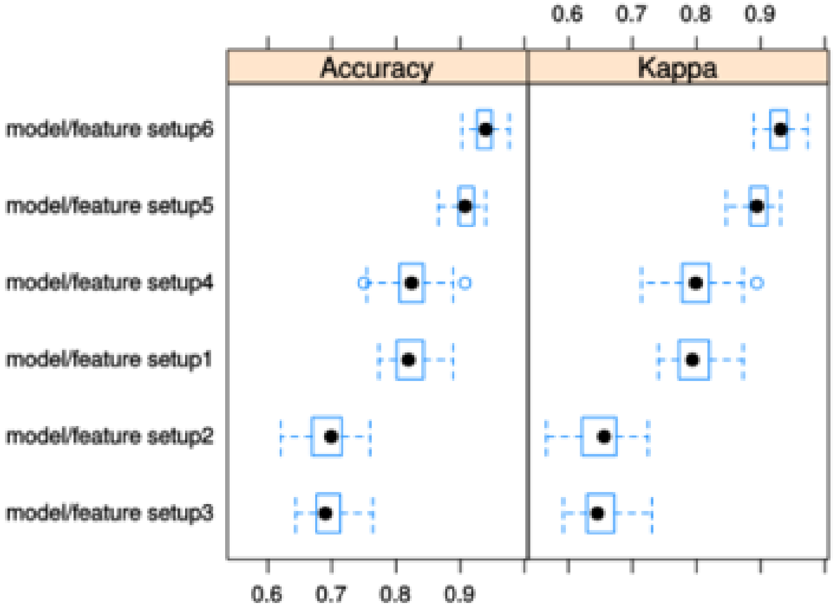


Gesture: "triangle"

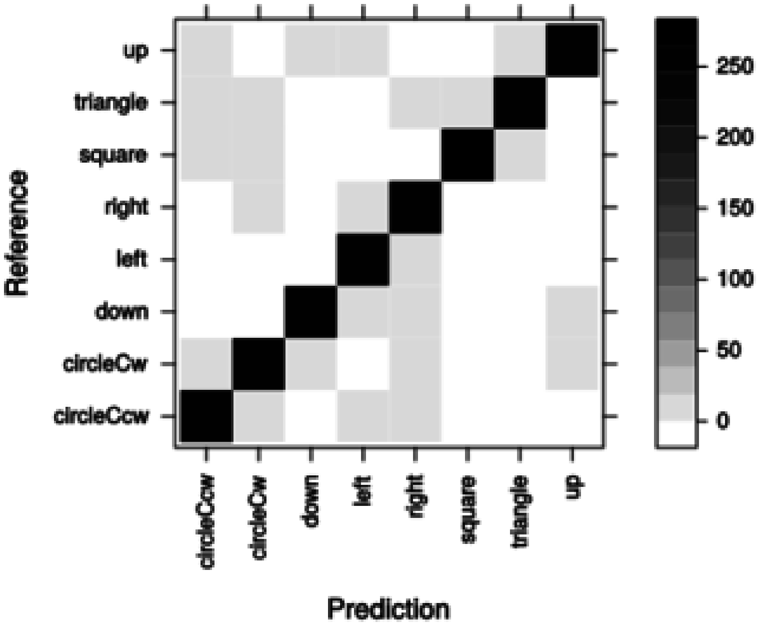
Example features (excerpt):



Possible model performances:



Possible confusion matrix:


Reference
up
triangle
square
right
left
down
circleCw
circleCcw
Prediction
circleCcw
circleCw
down
left
right
square
triangle
up
250
200
150
100
50
0