## horizontal line



CSE 572 Data Mining Assignment 3  
GROUP 9

10/23/2017

Alankrit Shah

Nikhil Lohia

Srinivas Murali

Vidhi Patel

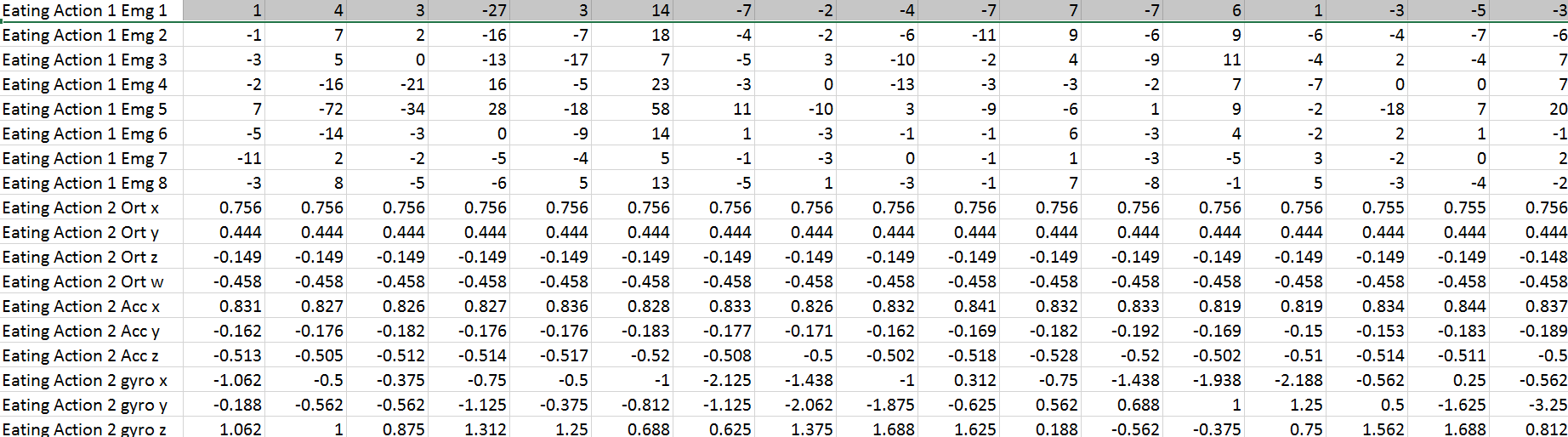
# Overview

The project involves feature extraction and feature selection from raw sensor data collected during the previous assignment during eating of food. In the raw data, there are 17 data streams: a) 3 from accelerometer, b) 3 from gyroscope, c) 3 from orientation, and d) 8 from EMG sensors.

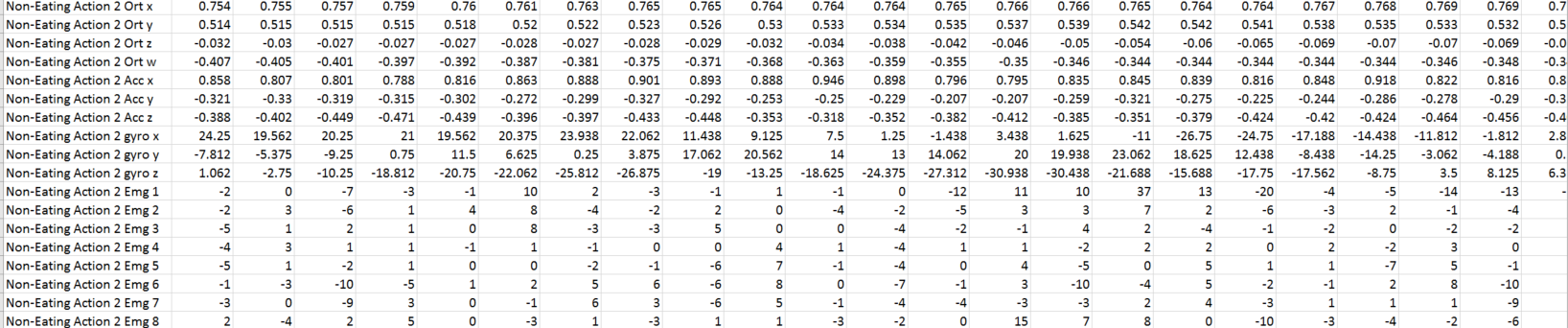
The project is divided into 3 major tasks and several subtasks. Here are the tasks and their explanations based on the results of the analysis done in MATLAB:

Task 1:

We segmented the annotations collected during Assignment-2 into two separate classes: Eating and Non-eating. We have 2 csv files for the 2 classes with 18 rows of each sensor values with time series of each action stored column-wise.



*Img1: Data for eating action*



*Img2: Data for non-eating action*

The following is our folder structure:

1. Annotations : Contains annotations from all the groups. We cleaned and make them into common structured files since all of them had separate structures.
2. EMG: Contains all the EMG data
3. IMU: Contains all IMU data
4. Eat.csv and non.csv: Contains eating and non-eating actions data
5. *Task1.m*: Contains MATLAB code for the task 1

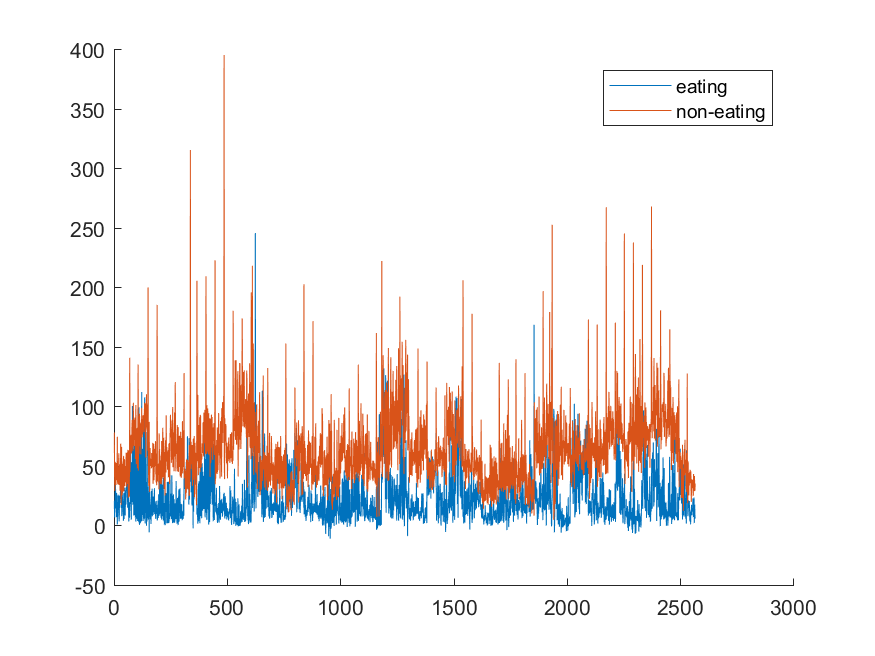
Task 2:

Intuitions for the features chosen:

1. Range of all EMG values
2. Acc\_x: When eating, x is mostly constant whereas while non-eating, x is variable since we are switching between sections
3. Acc\_z: While eating, z is changing whereas while non-eating, z is almost constant
4. All Gyro values give good difference between eating and non-eating actions because during a non-eating action, the hand will not rotate. We will be able to see some kind of rotations when we are eating the food.
5. Both MIN and MAX are good measures of feature extraction but they work upon the same underlying idea. We choose only one of them.
6. Standard Deviation also gives a good distinction between eating and non-eating actions, especially when considering orthogonal values

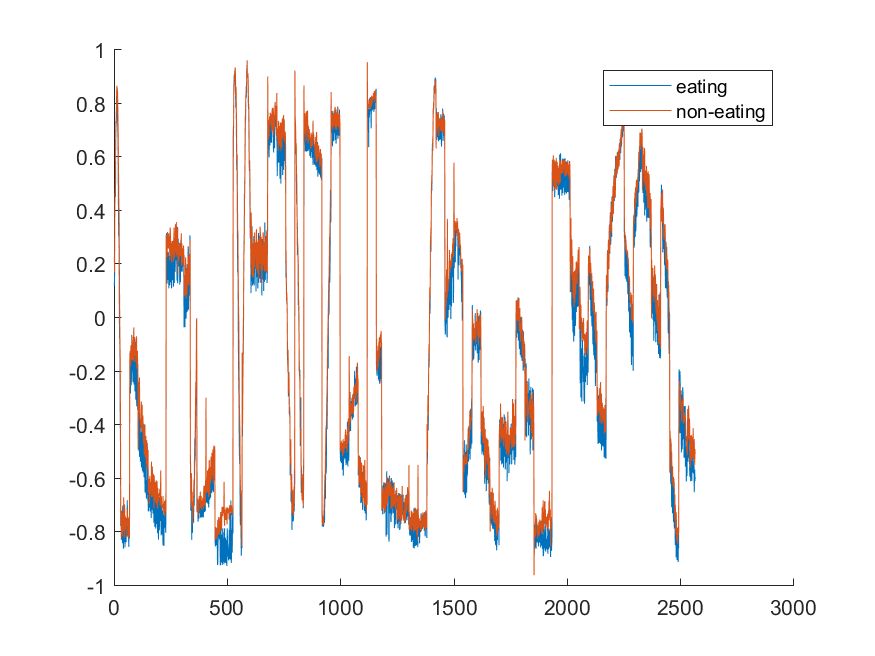
Methods for feature extraction:

1. **MAX**
   1. The MAX suggests the maximum value out of a time series
   2. MAX is a good choice for feature extraction because when it has a range of values when eating and non-eating. This can help to distinguish between the two classes.
   3. MATLAB code is present in the file ***<FeatureExtraction.m>***



*Max for Gyroscopic Y*

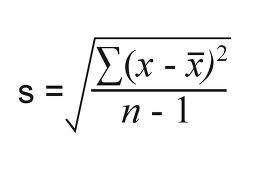
* 1. While eating, the gyroscopic values in Y direction tend to be lesser than for eating action. The intuition behind it is a person tends to move or rotate his/her hand more in the lateral direction while picking up food or just moving the spoon around without carrying the food to the mouth.

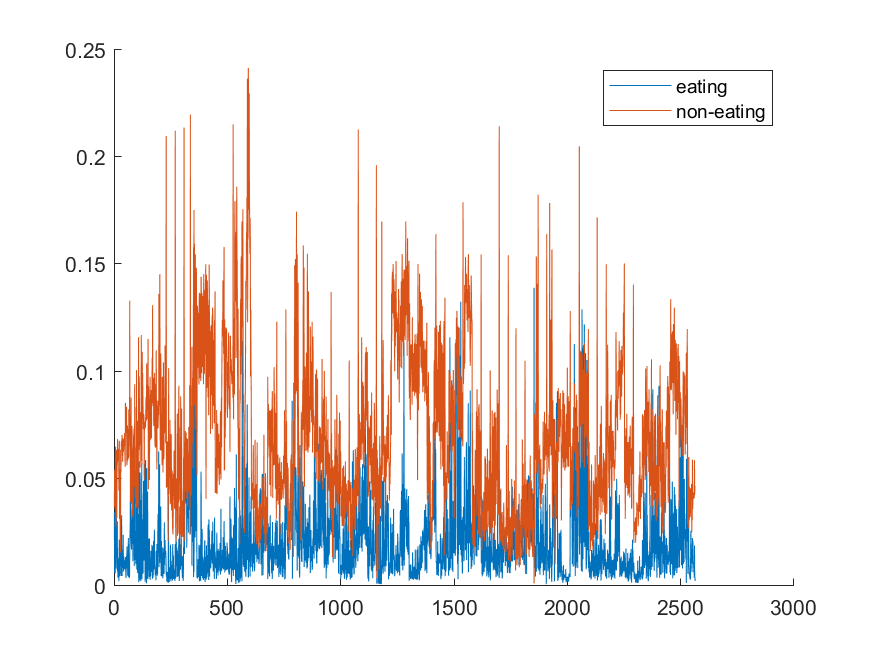


*Max for Gyroscopic Y*

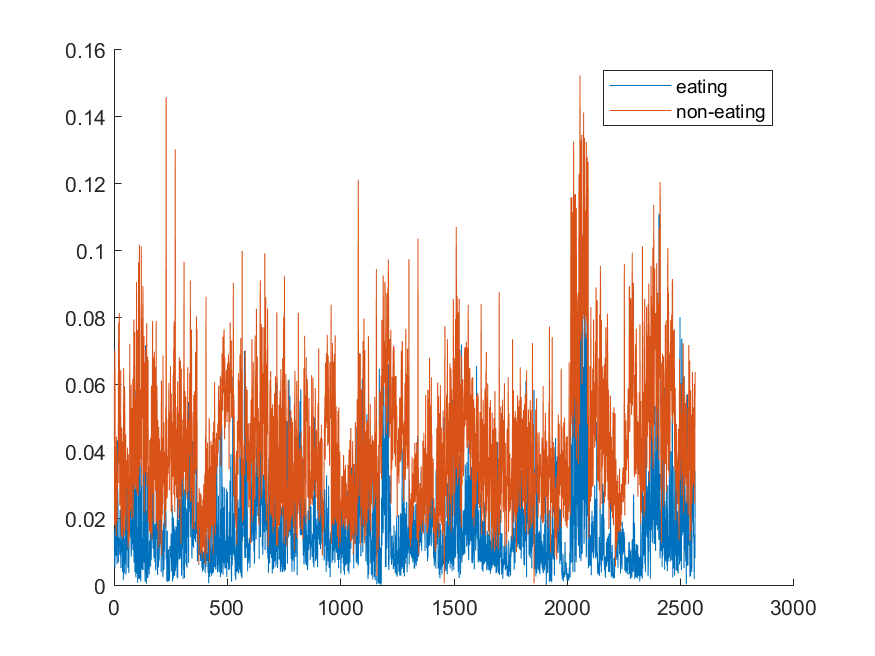
Max values does not work in situations where the acceleration goes in equal magnitude in both positive and negative directions. For examples, MAX value for Acceleration X is not a good feature to distinguish between eating and non-eating actions.

**This means that our original intuition was correct.**

1. **Standard Deviation**
   1. Standard deviation for a time series is calculated as  
      
   2. SD gives an idea about how much does the values vary with each other.  
      An eating action comprises of picking up the spoon and carrying it to the mouth.  
      A non-eating action involves activities such as chewing of food, fiddling with the spoon or simply keeping the spoon steady.  
      Hence, there would be a variation in a range of time series which is captured by SD.
   3. MATLAB code is present in the file ***<FeatureExtraction.m>***

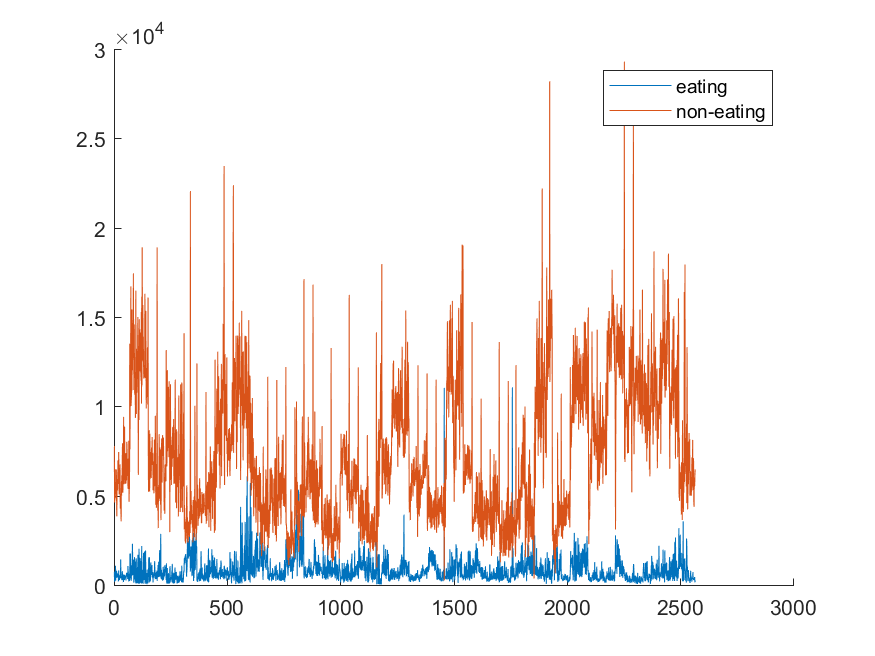


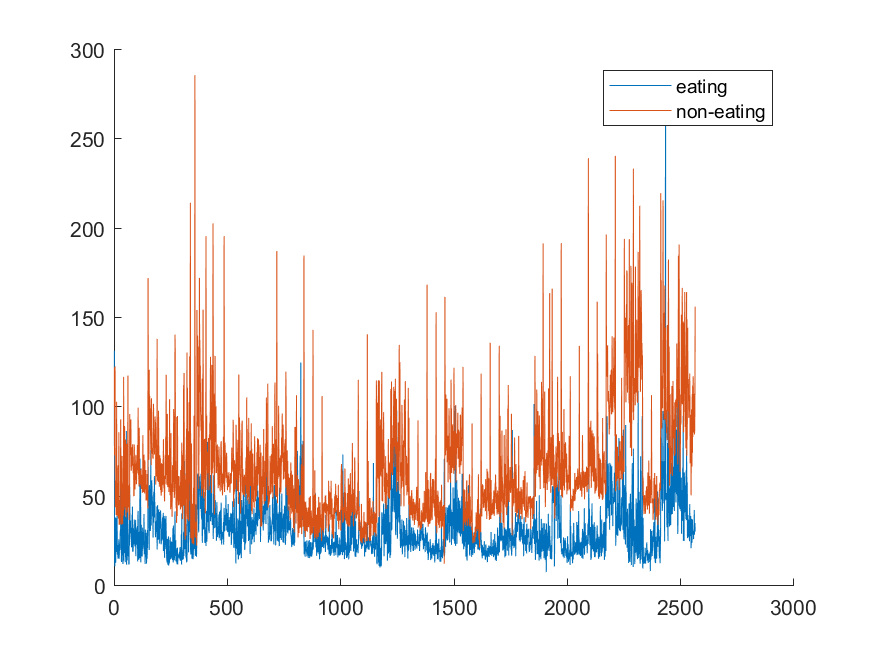
*Standard Deviation for Orientation Z*

**

*Standard Deviation for Acceleration X*

* 1. For non-eating action, a lot of variation is observed due to motion of hand which gives a good standard deviation in orientation feature in Z direction. Inversely, for eating action, there is less variation in orientation due to less fidgeting of hand while carrying food to the mouth.

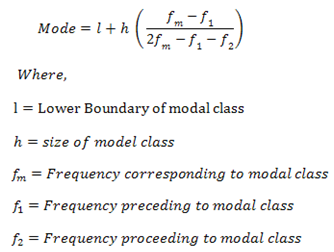
1. **Fast Fourier Transformation**
   1. Fourier analysis converts a signal from its original domain (often time or space) to a representation in the frequency domain and vice versa. An FFT rapidly computes such transformations by factorizing the DFT matrix into a product of sparse (mostly zero) factors. We used the in-built MATLAB function to calculate FFT.  
        
      *Y = fft(X,n,dim) returns the Fourier transform along the dimension dim. For example, if X is a matrix, then fft(X,n,2) returns the n-point Fourier transform of each row.*
   2. FFT works upon the underlying idea to capture the most dominant frequency in time-series data. The eating action and non-eating action have been annotated as a time series. Hence, this would help us in figuring out the class.
   3. MATLAB code is present in the file ***<FeatureExtraction.m>***
   4.   
      *FFT for Gyroscopic Y*
   5. Both the eating and non-eating actions are a cycle of actions and hence their frequency of signals would repeat again and again. The FFT hence is a good method to capture the dominant frequencies for different features. It worked out for a lot of features in the data.
2. **Discrete Wavelet Transform**
   1. In numerical analysis and functional analysis, a discrete wavelet transform (DWT) is any wavelet transform for which the wavelets are discretely sampled. As with other wavelet transforms, a key advantage it has over Fourier transforms is temporal resolution: it captures both frequency and location information (location in time).
   2. Similar to FFT, DWT captures both the frequency and the location information of time series data that would help us in classifying data as either eating or non-eating.
   3. MATLAB code is present in the file ***<FeatureExtraction.m>***



*DWT for EMG*

* 1. The EMG data captures the hand movements in a cycle of motion. Since both eating and non-eating actions have the same motions for each cycle, DWT of EMG data gives a clear distinction between the eating and non-eating actions.

1. **Mode**
   1. A Mode is calculated as the maximum frequency of an element in the time series data

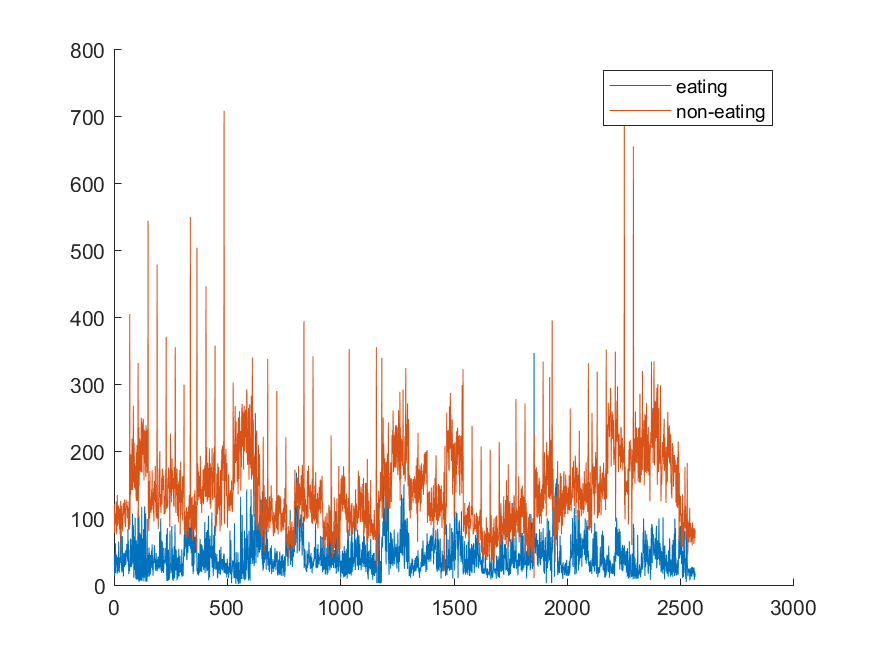


* 1. While observing the raw data for EMG, we observed that the frequency of some values were distinct in eating and non-eating action data. Hence, mode would give us a clear distinction between eating and non-eating actions.
  2. MATLAB code is present in the file ***<FeatureExtraction.m>***

|  |  |  |
| --- | --- | --- |
|  |  |  |

* 1. **The MODE values for none of the features worked out. Hence, our initial intuition about MODE was incorrect.**

1. **Range**
   1. The Range is the difference between the lowest and highest values.
   2. While eating, the range of accelerometer is more compared to a non-eating range of values. The reason for this is it is by intuition, while putting the spoon down, we tend to put it down faster than while carrying food. Hence, range can be used to find out the eating or non-eating action.
   3. MATLAB code is present in the file ***<FeatureExtraction.m>***



*Range for Gyroscopic Y*

* 1. Range for features such as Gyroscopic X, Y, Z and accelerometer etc. gave us a pretty stark difference between eating and non-eating set of values. The reason being, for all of these features, the values would tend to vary between a certain range, which gives us a good distinction for FFT and DWT as well. Hence, Range is a good parameter to judge an eating and non-eating values.

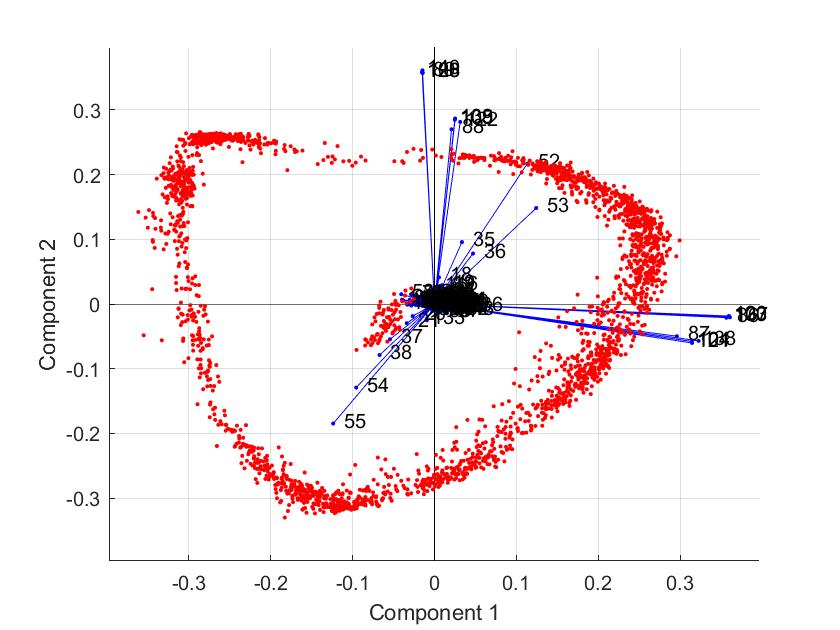
Task 3:

**Subtask 1:**

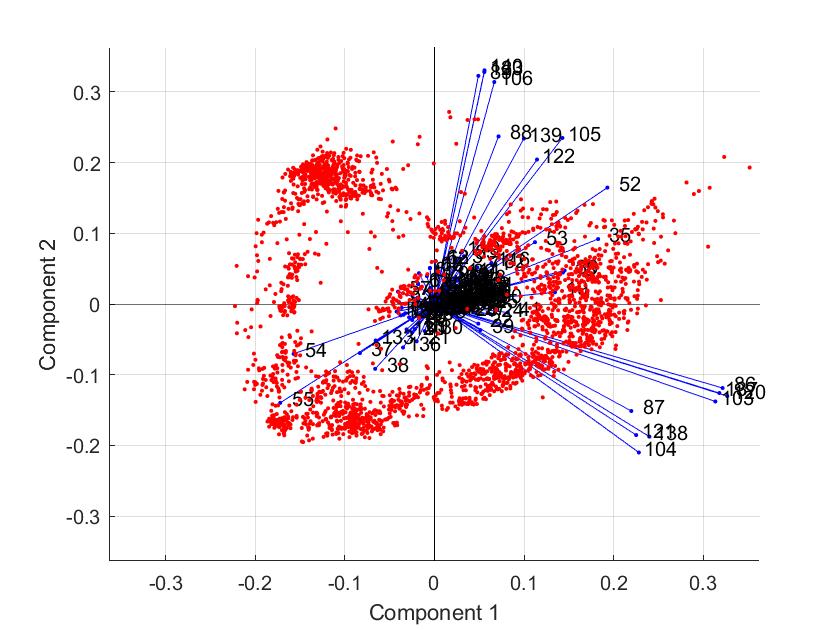
* To filter out contrasting values we plotted the eating and non eating action graphs for all the feature extraction methods(ex. max, min, standard deviation, rms, mode, range).
* Initially we had more than 150 graphs out of which we selected 12, which were visually different.
* We found standard deviation and range to be amongst the best feature selection methods and acceleration on z axis and gyro on y-axis to be few of the best features from all the features.

**Subtask 2:**

* We used the MATLAB’s built in *pca* function to calculate the principal component analysis for our feature matrix.
* We also generated 2 eigenvector plots, which show the constituent signals which contribute to the PCAs from eating and non-eating categories.



Eigenvectors of eating actions



Eigenvectors of non-eating actions

**Subtask 3:**

* The eigenvectors show the contributing features that construct the principal component.
* The length or magnitude of the eigenvectors are the corresponding eigenvalues which relate to the extent to which they affect the principal components.

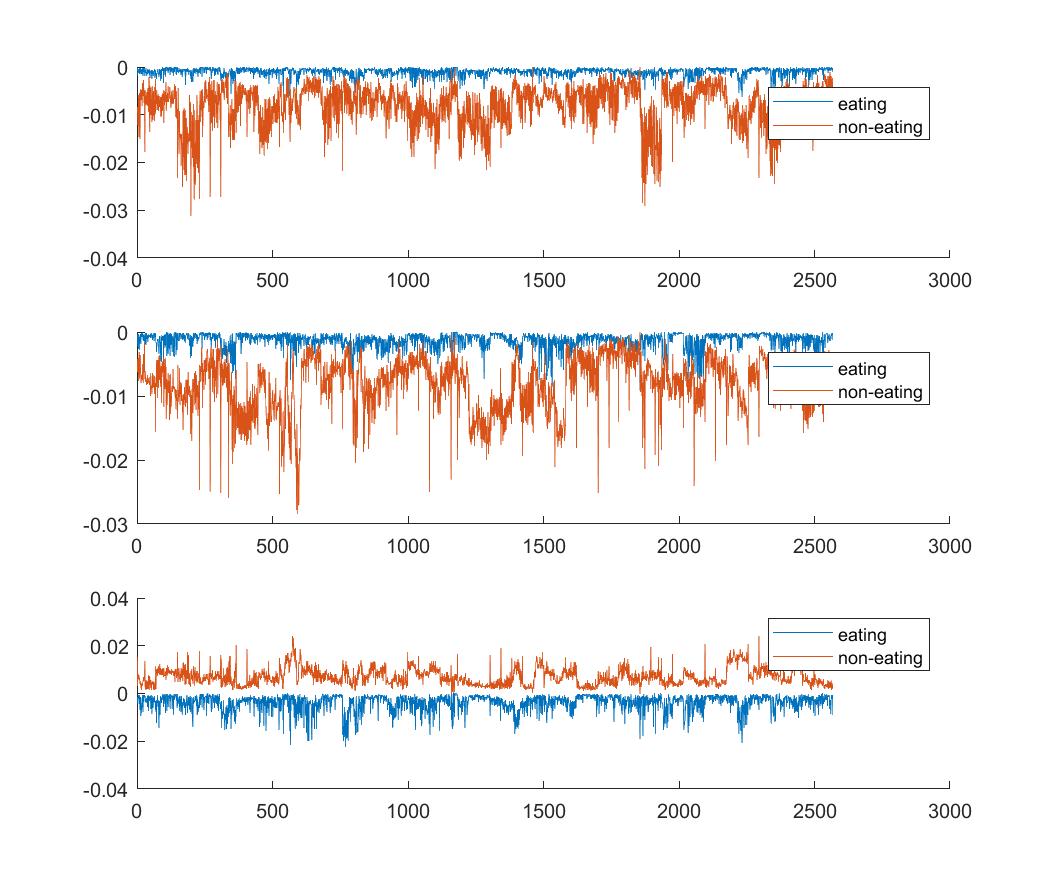
**Subtask 4:**

* To generate the new feature matrix, we multiply the output of the PCA to the original matrix to transform it to new feature space:

Coeff = PCA(OriginalFeatureMatrix)

NewFeatureMatrix = OriginalFeatureMatrix \* Coeff

* The figure below shows the 3 primary principal components.
* Each one of them shows distinction between eating and non eating action.

**Subtask 5**:

* PCA was helpful in reducing the dimensionality of the feature matrix.
* It also helped us understand and verify which features contributed the most for making the distinction between eating and non eating action.

# References:

* <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2936372/>
* <https://en.wikipedia.org/wiki/Fast_Fourier_transform>
* https://en.wikipedia.org/wiki/Discrete\_wavelet\_transform