CSE 572 Data Mining

Activity Recognition using Myo Sensor

Arizona State University

GROUP NUMBER: 14

Group Members:

Simran Singh	1214917639
Pratik Sahoo	1215063460
Krishna Babuji	1215039332
Krithika Mahalingam	1215193486
Shubham Vipul Majmudar	1215200298

1 Introduction

Activity recognition task is a growing field now, with many applications in our daily lives. It has been shown to be particularly effective in early medical diagnoses and for rehabilitation of patients and accident victims. In this project, we aim to develop a computing system that identifies human activities. We use a myo band for capturing the various sensor data pertaining to the arm movements during activities like eating, typing, sleeping. Then we apply data mining techniques to distinguish between between the activities.

1.1 Keywords

Feature processing, Feature extraction, Dimension Reductionality

1.2 Goal Description

There are three phases for this :

Phase 1 : Collect the activity data for various activities throughout a day by wearing the Myo sensor and retrieving the data.

Phase 2 : Perform feature extraction from the data collected in phase 1.

Phase 3 : Select features that are highly discriminative for each of the activities selected in Phase 1.

1.3 Assumptions

1. Data does not contain any outliers that can influence the outcomes
2. Mean, standard deviation, min, and max of the episodes are very useful in identifying each activity.

1.4 Terminology

- EMG : Electromyography
- IMU : Inertial Measurement Unit
- RMS : Root Mean Square
- PCA : Principal Component Analysis

2 Description of the Proposed Solution

2.1 Phase 1

The Myo armband measures electrical activity from the muscles using EMG sensors to detect five gestures made by the hands. Using a 9-axis IMU, it also senses the motion, orientation and rotation of your forearm. We received the sensors for 2 days. It was worn by two members of team in order to record the activities. Data for the following activities were recorded along with the timestamps.

- Eating : contains a series of movements in wrist
- Sleeping : very minimal movement

The Myo armband transmits this information over a Bluetooth Smart connection to communicate with compatible devices. Myo also streams the raw EMG and IMU data. We got the data in 5 csv files for

1. accelerometer,
2. emg,
3. gyro,
4. orientation and
5. orientation euler.

Hardware

The Myo armband has a medical grade stainless steel EMG sensors, highly sensitive nine-axis IMU containing three-axis gyroscope, three-axis accelerometer, four-axis orientation sensor. The Myo is able to detect any tiny electrical activation that the muscle produce. Myo armband is developed by Thalmic labs and we got the technical specifications from their website for understanding the hardware.

2.2 Phase 2

(a) The number of instances for the action 'Eat Food1' are roughly 15,000 and there are several parameters, each with multiple features for each instance. If we consider this entire matrix, the dimensionality and the cost of computation is unrealistically high for practical purposes.

We are hence only considering a set of instances rather than every instance in order to further reduce the complexity, and the features corresponding to this window of instances are condensed from the original features by using statistical metrics. Based on intuition, only features that are found relevant (i.e. features that should clearly indicate different values for the two labels- eating and not eating) are being extracted. We will further validate these assumptions for every metric by creating various plottings and attempting to distinguish them visually.

The statistical metrics taken into consideration are mean, minimum, maximum, standard deviation and root mean square (we did not use the Fast Fourier Transform because it gives inaccurate output for an input signal that is not of the form 2^n and also not a whole numbered cycle in that range- which is a very common case with signals). These metrics are applied to the features of a window of 64 instances (in case of IMU parameters), resulting in a total of 225 windows, each consisting the aforementioned statistical features for each parameter. To be completely accurate, since we want instances of one eating action and we have several such instances present in this activity label, one window size should be set as one instance of an eating action. This is a pattern recognition problem, more specifically, that of identifying recurring patterns in the signal so as to set a dynamic window size that takes the range of one eating action. However, we were not able to figure this out with certainty and hence assigned an arbitrary, constant window size. The frequency of the band for capturing the IMU parameters is 50 Hz, whereas the frequency for EMG is 200 Hz. Hence for every IMU data point collected, there are 4 data points corresponding to EMG collected within that period. Thus the window size for EMG is set to 64 times 4, which is 256.

(b) The Accelerometer measures the acceleration with respect to the hand movement when a particular activity is performed. Eating involves a series of hand movements resulting in some acceleration such as:

a) Picking up food: Requires some hand movements in the horizontal direction.

b) Raising the spoon: Requires some hand movements in the vertical direction against the gravity hence resulting in a positive acceleration.

c) Placing food in the mouth: Senses some negative acceleration as it goes back to ‘g’ (gravitational acceleration) from a positive value which was applied to bring the food upwards.

d) Bringing the spoon down and repeat: Requires movement in downward direction along the gravity hence resulting in an acceleration of a magnitude of at least 9.8 m/s^2 in the negative direction.

Hence we can observe variations in the acceleration throughout the eating activity resulting in extreme peaks in the signal. Whereas if there is no activity, there would be minimal movement and as a result acceleration would not have much variation. Since there is observation of considerable variation in the plots for eating and non-eating activity, it is in our best interest to take these features for extraction as the acceleration involves negative minimum and positive maximum values.

Minimum: The minimum values for eating should differ considerably with the respective minimum values for non-eating activity since eating involves many hand movements.

Maximum: Considering the maximum values of activities will give us peak values in a higher range for eating activity as compared to non eating activity, where maximum values of non eating activity (in our case no movement) will have peak values at lower range due to less variation.

Average: Taking Average of activities values should give us distinct patterns as average of eating activities will fall in a different range of values with respect to average of non eating activity.

Standard Deviation: The Standard deviation of eating activity values will be more varied as compared to the standard deviation values of non eating activity. Hence considering Standard Deviation should provide us distinct pattern for eating vs non eating.

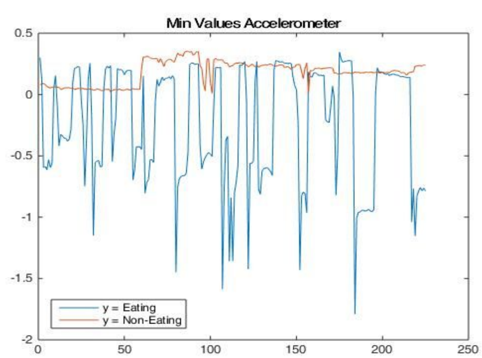
Root Mean Square: Similar to average, taking RMS of activities values should give us distinct patterns as RMS of eating activities should fall in a different range of values with more variation with respect to RMS of non eating activity.

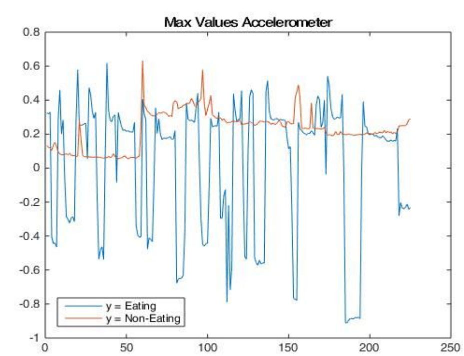
(c) The MATLAB code for extracting the features has been provided separately under the Code folder.

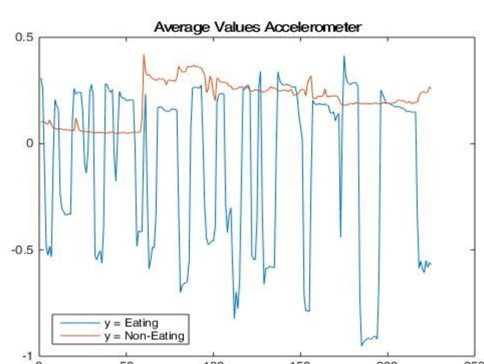
(d) Generating Plots corresponding to each activity

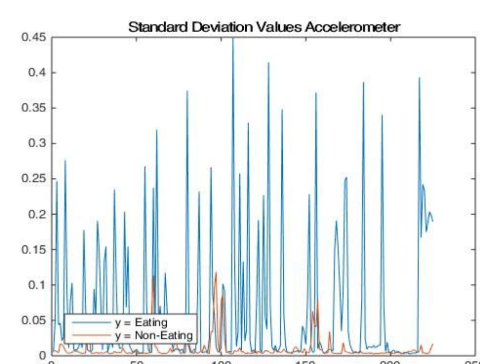
In this step we generated plots of eating and non-eating activity, where we overlapped the eating and non-eating trends based on the new features extracted i.e Minimum, Maximum, Average, Standard Deviation and RMS.

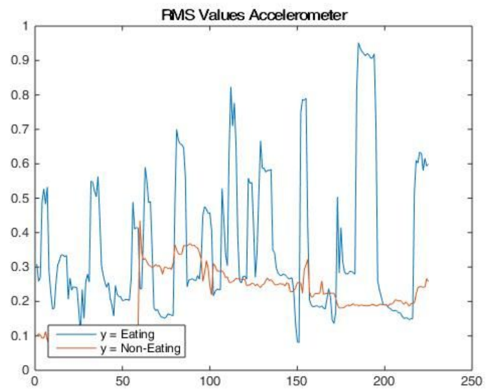
Plots where eating and non eating activities are clearly distinguishable.

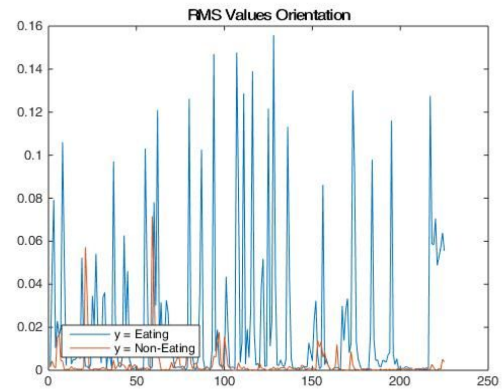




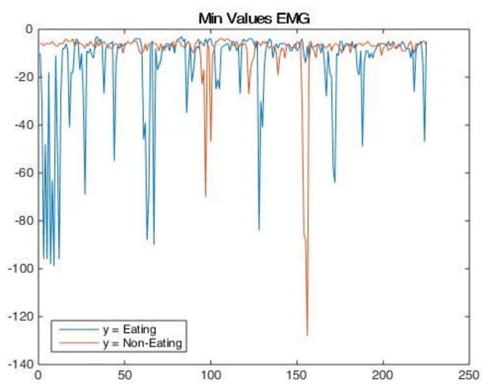


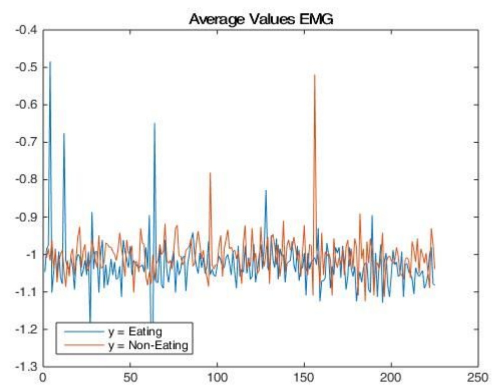






Plots where eating and non eating activities are not clearly distinguishable.





(e) Our initial intuition of selecting the parameters holds true for the following parameters: *accelerometer_min, accelerometer_max, accelerometer_mean, accelerometer_std, orientation_rms and accelerometer_rms*. *
However, when the plots were made and analyzed, our initial intuition of selecting the parameters did not hold true for the following parameters:
emg_mean and emg_min.

2.3 Phase 3

The data matrix is arranged such that the instances are placed in rows and the columns contain the statistical features obtained from the various parameters all laid down side-to-side. This makes the data matrix of size 225 x 105 (number of instances/windows X number of features) for each activity- eating and non-eating.

Now that the input matrix is ready, we can apply PCA to each of them. The default PCA technique used in MATLAB is Singular Value Decomposition, which requires the matrix to be centered before taking its covariance. Fortunately, MATLAB does this for us before applying PCA and we get the output values in the following order

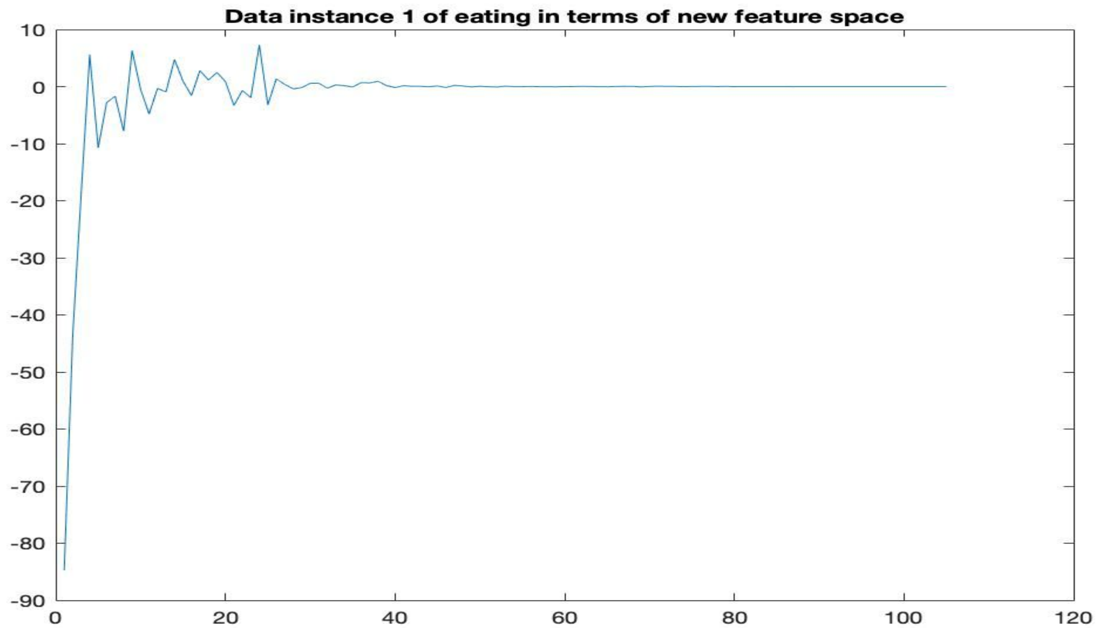
- coefficient of the PC's, meaning the loadings for each principle component. These indicate the contribution of the original variables on the new features,
- PCA score, i.e. the projection of the data on the new feature space, and
- Latent matrix, which indicates the principle component variances.

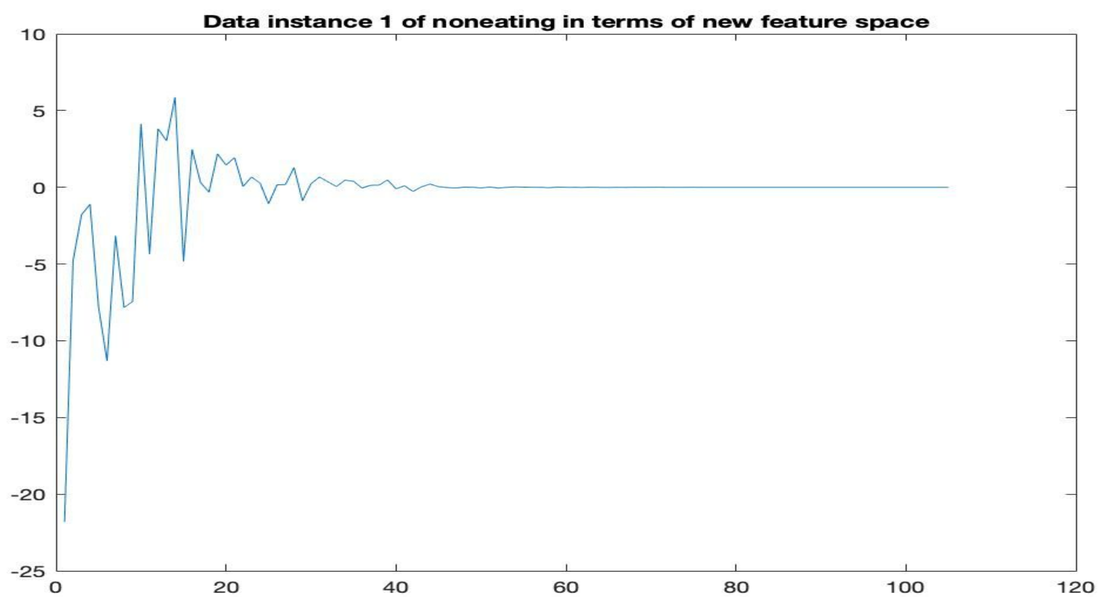
The corresponding outputs are given in descending order of the latent matrix, indicating the most similar principle component in the first column and the others gradually in decreasing order of importance.

In our case, PCA is performed for each of the two labels- 'eating' and 'non-eating'. This, hence results in these two data matrices being represented in two different basis vector spaces. If we would like our activities to be present in the same basis vector space, we would be required to apply the PCA to the combined data matrix of activity and non-activity. Note that this retains the number of features, and only increases the number of instances.

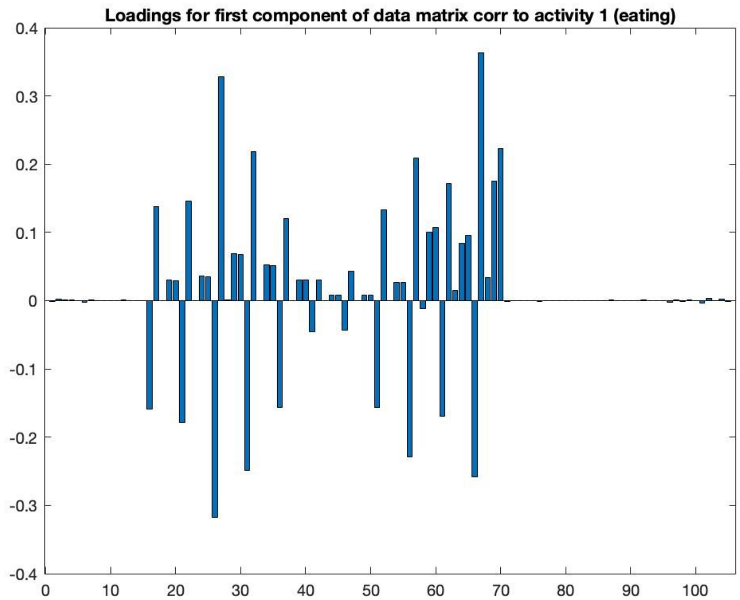
The new feature matrix for class 'Eating' is obtained by multiplying its original data matrix (centered) with the principle components. This is done by the *pca() function* in MATLAB and stored in the second output variable, namely 'score' (since the projections of the data matrix on the new feature space are also called PC scores).
The same output is obtained for when we perform *pca()* on the matrix of the second activity class 'non-eating'. The structure of this output is that the columns are now the new features and the rows are the instances. A point to note is that this projection matrix has the same dimensions as the original data matrix; it is only after we select the features that we prune the columns from the end.

The figures below show the first instance of our data in terms of the new features. Note that the values corresponding to the features gradually taper, as is the case with other data points as well. This is indicative of the loss of variance as the features are added.

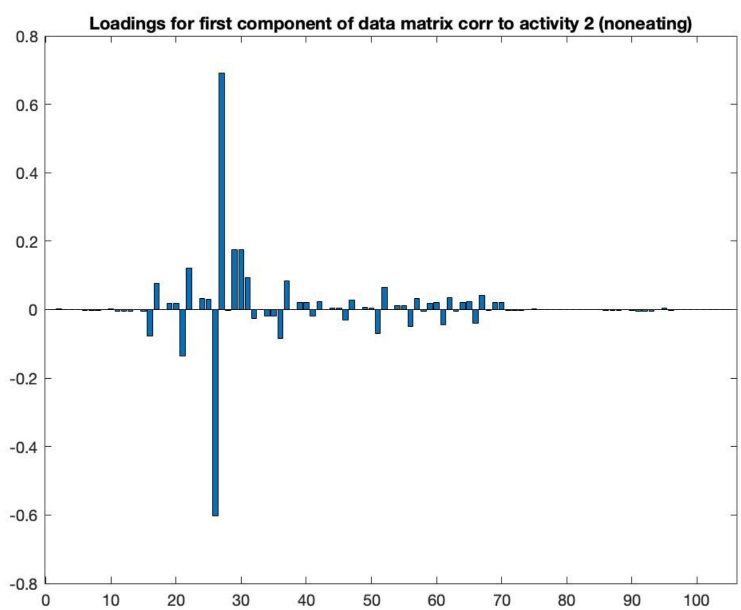


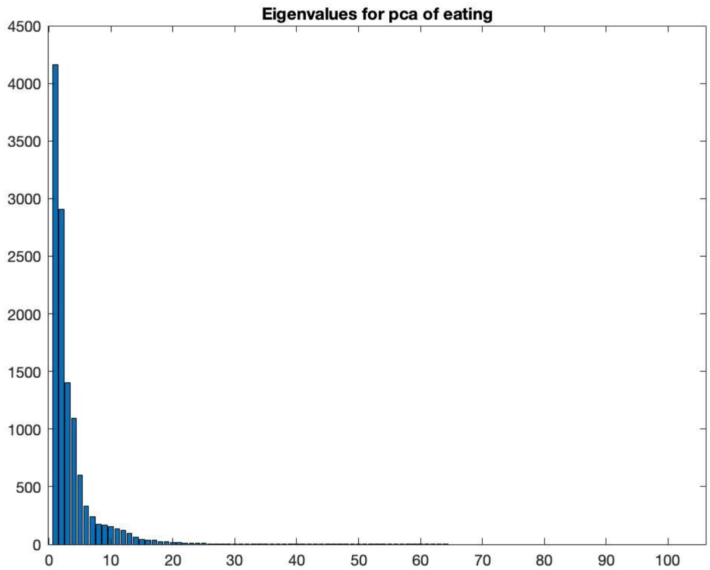


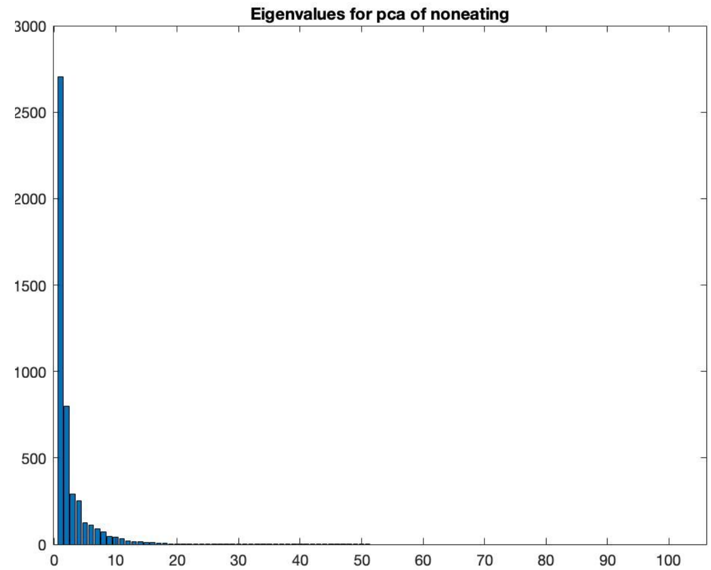
The eigenvectors are the unit scaled factor loadings for each principle component. These 'loadings' of each variable (original) to a particular feature are contained in an i x i matrix, which is the key to understanding the nature of a particular factor in terms of the original feature space.



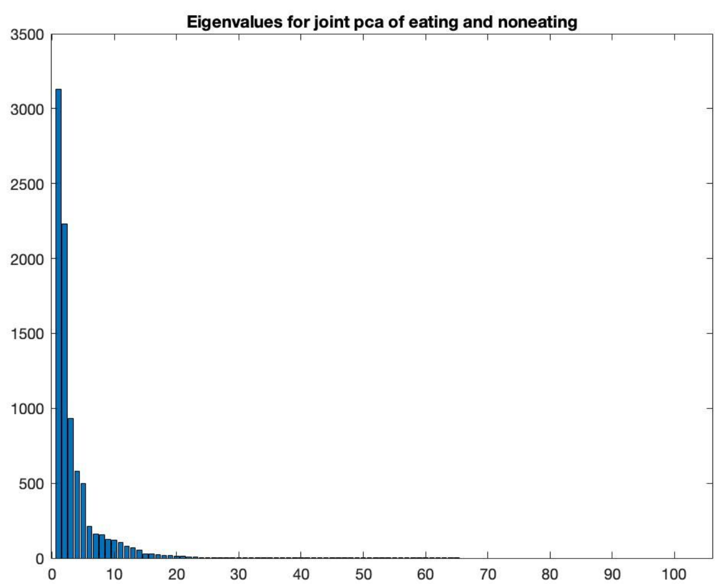
After observing the above bar graph, it is seen that the features corresponding to the min and max metrics for almost every given parameter contribute largely to the first principle component. For the second activity (non-eating), as expected there will not be much of a intraclass difference between the min and max values of the accelerometer, gyroscope, etc. Hence the only features majorly contributing to the new feature space are the min and max of the emg readings.



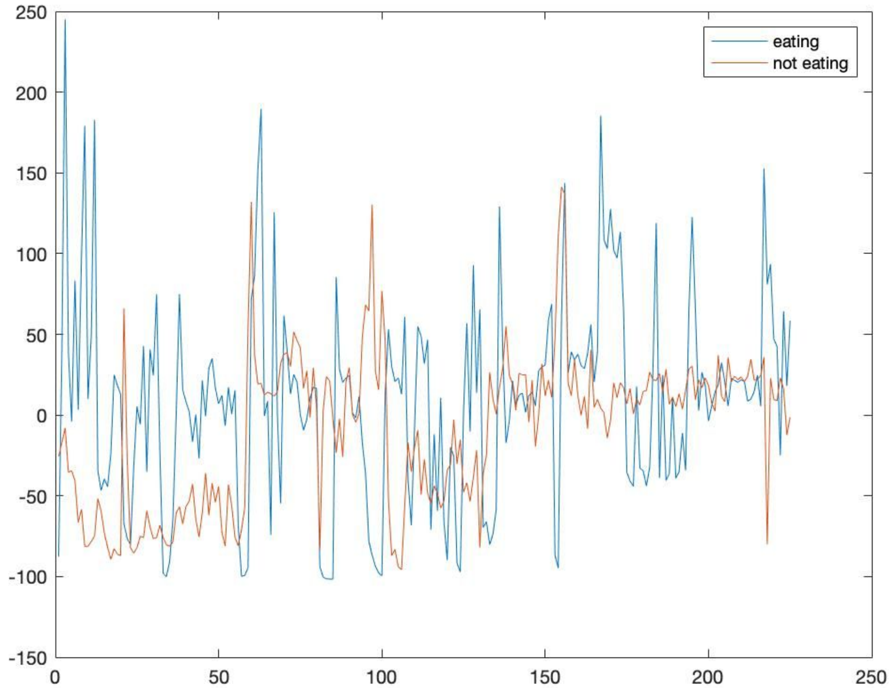




The eigenvalues of the pca saturate very quickly (approx. 95% variance for eating and 98% for non eating covered in 11 features), indicating that most of our original data was not very informative. We can safely, hence eliminate the remaining 94 features.



When we perform the PCA on the joint matrix of eating and non eating, we get similar results for percentage variance retained in 11 features (approx 95%).



When we need to use PCA for classifying a new data point into eating/non-eating, we need to make sure that all our data (eating and non eating) is represented in the same basis vector space. Thus there will only be one PCA performed on the joint eating and non eating data. The above diagram represents the first principal component score of each eating and non eating in the new vector space. Being the first PC, eating and non eating instances should be more distinguishable, but the reason behind such variance is as follows- there is a considerable intraclass variance in the first set 'eating' (for which we have considered 'Eating Food1'), whereas the second set 'non eating' (for which we have taken 'NoActivity') has a very low variance. When we perform PCA on their joint matrix, there is an increase in the overall variance and hence the data corresponding to non eating has also been affected in terms of variance (intraclass variance is increased) and a clear demarcation is not visible.

3 Conclusion

In conclusion, we were able to see identifiable patterns in Myo Sensor data that we collected for the activities that we chose and also verify some of the assumptions that we made in the beginning. Our analysis has confirmed that PCA is indeed helpful because we were able to preserve the 95% of the original variance even after dropping the number of features by a factor of 10; thus reducing the space and allowing classification algorithms to perform better.

4 References

[1] : Toutountzi, Theodora & Collander, Chris & Phan, Scott & Makedon, Fillia. (2016). EyeOn: An Activity Recognition System using MYO Armband. 1-3. 10.1145/2910674.2910687

[2] : Mathworks pca() documentation