Activity Recognition Project

Group #1

Adric Rukkila

Jhe-Yu Liou

Stephanie Bienz

Tristan Le

Uttam Chowdary Jagarlamudi

CSE 572: Data Mining
Fall 2018

Table of Contents

Phase One: Data Collection

Two out the five group members wore the Myo wristband on their dominant arm for two and a half days. The times where special activities occurred (meals and otherwise) were logged by each member, which can be seen in the table below.

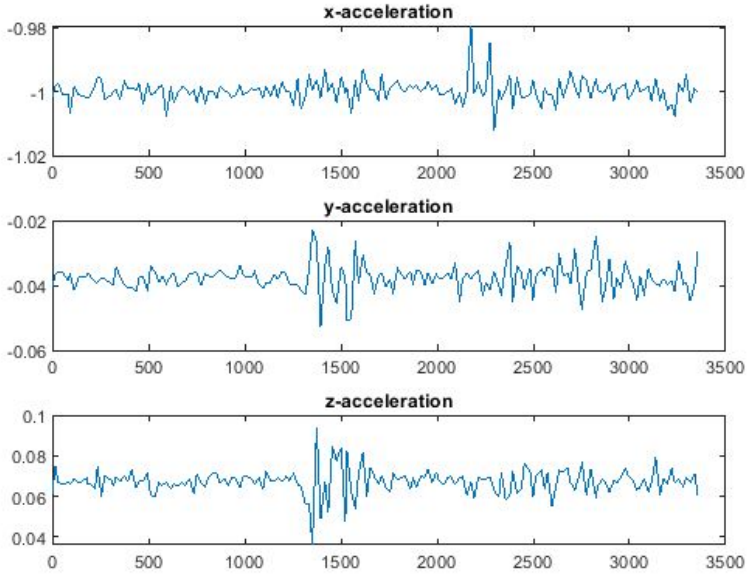
Date	**Start Time**	**End Time**	**Activity**
September 14, 2018	6:42 PM	6:58 PM	Eating with bowl and chopsticks
September 14, 2018	7:41 PM	7:48 PM	Driving
September 14, 2018	8:10 PM	8:27 PM	Playing badminton
September 15, 2018	10:38 AM	10:50 AM	Eating with chopsticks
September 15, 2018	1:30 PM	2:40 PM	Working out
September 15, 2018	3:36 PM	4:04 PM	Eating KFC
September 16, 2018	9:25 AM	9:30 AM	Driving
September 16, 2018	7:15 PM	7:26 PM	Eating Pizza
September 16, 2018	8:40 PM	8:42 PM	Going up and down stairs

We decided to log more than one extra interesting activity (besides eating) in the event that the data collected for certain activities is not worth using or is uninteresting.

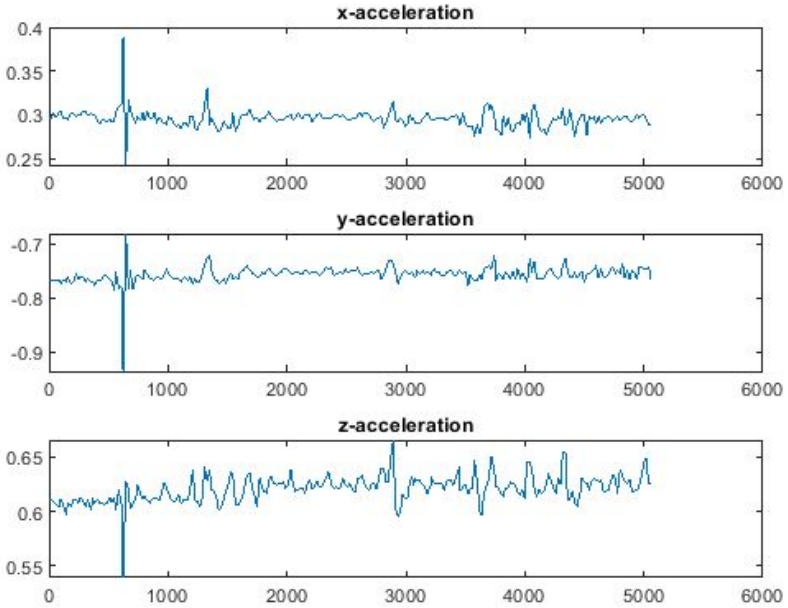
Phase Two: Feature Extraction

1. Fast Fourier Transform

Fast Fourier transform is a mathematical method for transforming a function of time into a function of frequency. The FFT is a fast algorithm for computing the DFT, where we take 2-point DFT and 4-point DFT and generalize them to 2^n-point DFT. We used this feature as it allows us to take a signal containing noise and decompose the signal frequencies of interest. For example, here is the acceleration data for the eat1 dataset:



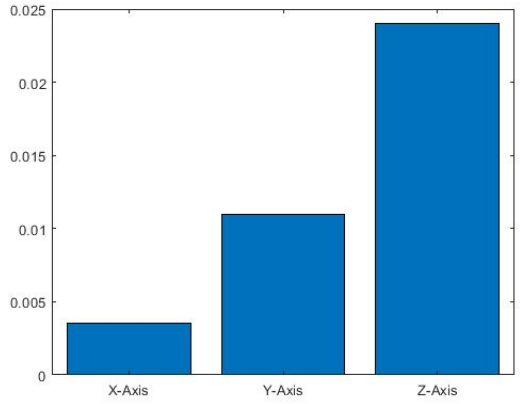
Whereas here is the acceleration data for the drive1 dataset:



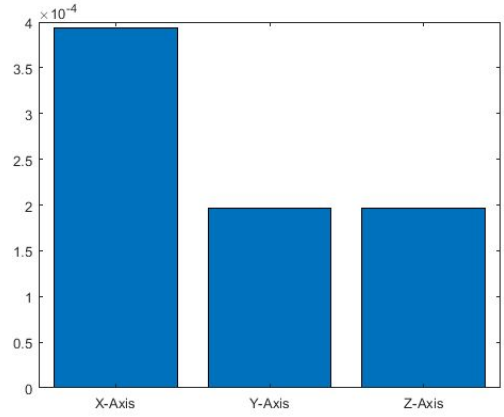
Our intuition is that the frequencies of these two datasets, and all eating vs. non-eating datasets, differs, and so FFT is a good feature to extract. We extract the peak frequency among the frequency spectrum.

Continuing the example, looking at the FFT results for acceleration showed this:

For the eat1 dataset:



For the drive1 dataset:

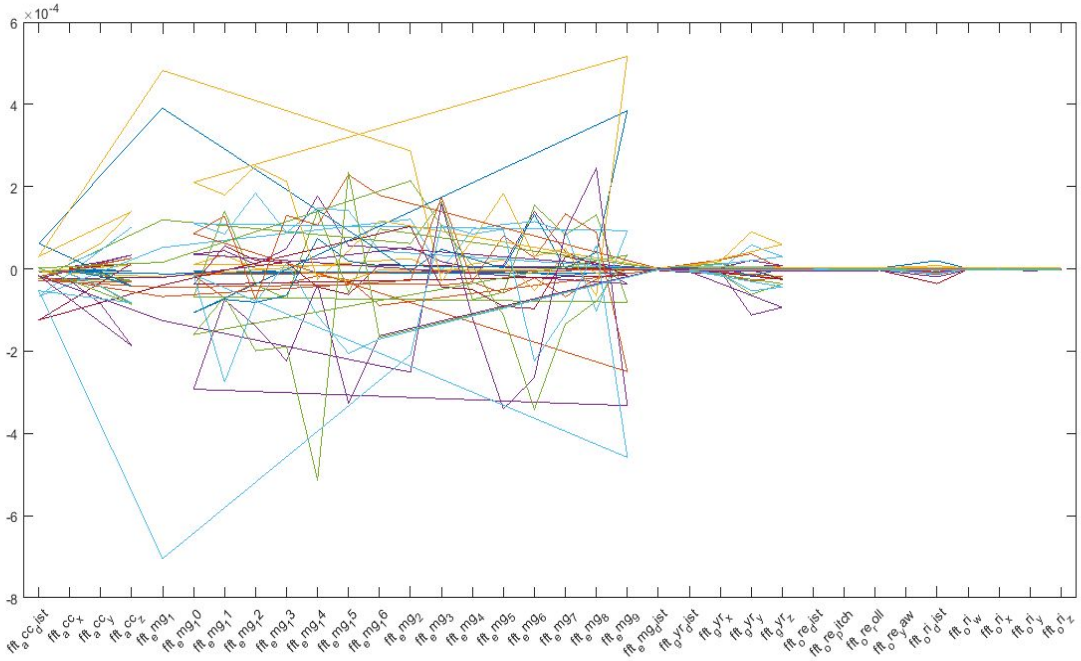


These appear to be significantly distinct X, Y, and Z FFT values. To confirm whether our intuition about acceleration was actually correct or not, though, we used PCA to compare the FFT of the eat1 and drive1 datasets:

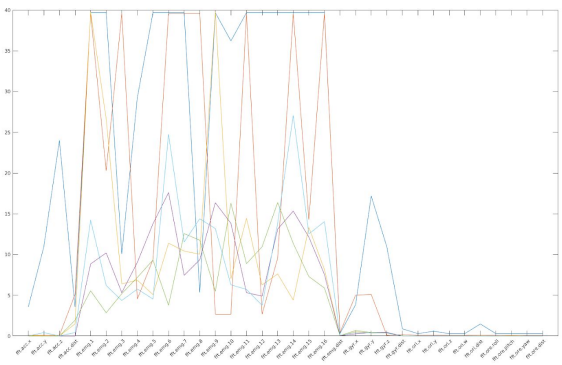
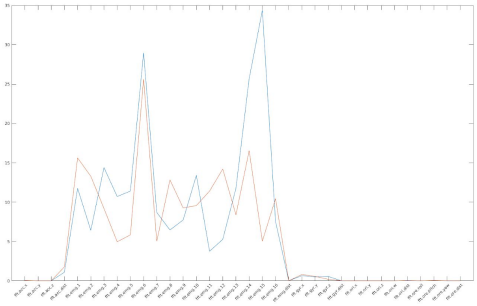
Feature	**Coefficient**	**Eigenvalue**
FFT of X-Axis Acceleration	0.016042	1.942807324742859e+04
FFT of Y-Axis Acceleration	0.054621	1.942807324742859e+04
FFT of Z-Axis Acceleration	0.12076	1.942807324742859e+04

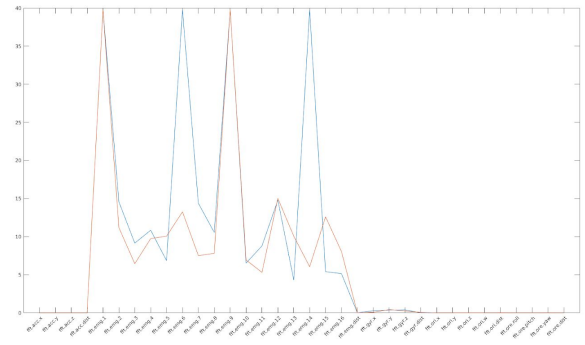
This tells us that while the FFT of Z-axis acceleration has a high degree of variance between eating and driving, the X- and Y-axis accelerations are much lower. So our initial intuition was partly confirmed and partly disproven.

We then repeated this example process of comparing eating vs. non-eating for all datasets:



Keeping in mind that FFT is good for EMG data. Breaking the data down further:

Eating FFT	Cooking FFT
	

Keyboard FFT	No Movement FFT
	

Using the threshold of the maximum value occurring in the top six eigenvector values, we reduced the full FFT featureset down to 8 features:

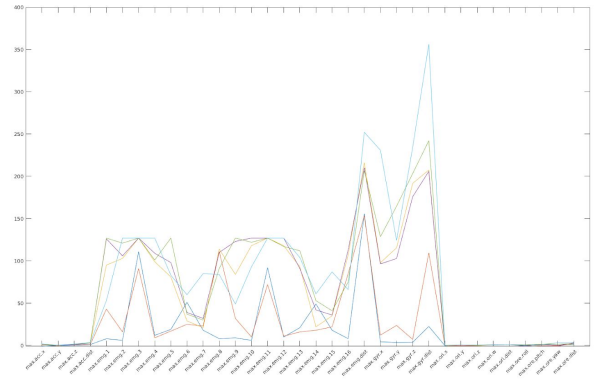
FFT Feature	**Coefficient**	**Column**	**Eigenvalue**
fft_acc_z	0.03444	4	2203.91909655853
fft_acc_dist	-0.045278	6	1529.83043728323
fft_emg_6	0.19633	4	2203.91909655853
fft_emg_8	0.12166	5	1922.4284847171
fft_emg_9	0.152	6	1529.83043728323
fft_emg_12	0.17041	6	1529.83043728323
fft_emg_14	0.22869	4	2203.91909655853
fft_gyr_y	-0.030338	2	8884.50659340742

2. Maximum

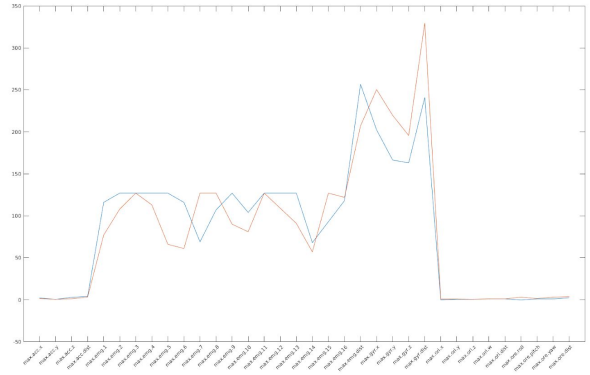
Utilizing maximum as a feature extraction tool is useful for seeing the distribution of high points or peaks on the various data points. This feature extraction is believed to be useful in distinguishing the stationary action and sporty action.

The following are the results of our maximum feature extraction for the various activities:

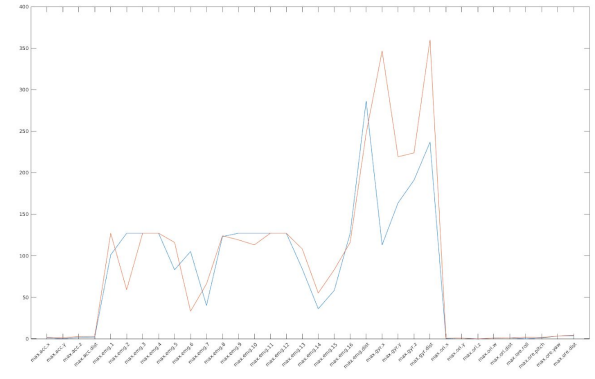
Eating MAX



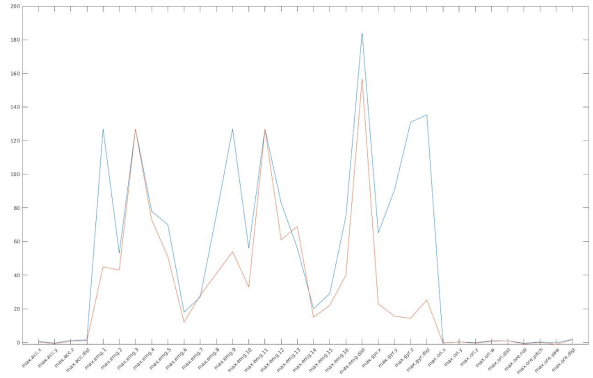
Cooking MAX



Keyboard MAX



No Movement MAX



We were then able to extract the eigenvalues from the data:

MAX Feature	**Coefficient**	**Column**	**Eigenvalue**
max_emg_2	0.30747	2	8884.50659340742
max_emg_4	0.27575	2	8884.50659340742
max_emg_6	0.41373	4	2203.91909655853
max_emg_8	0.49595	5	1922.4284847171
max_emg_9	0.33449	2	8884.50659340742
max_emg_12	0.32651	2	8884.50659340742
max_emg_15	-0.35492	3	3211.42693990932
max_emg_16	0.41592	5	1922.4284847171

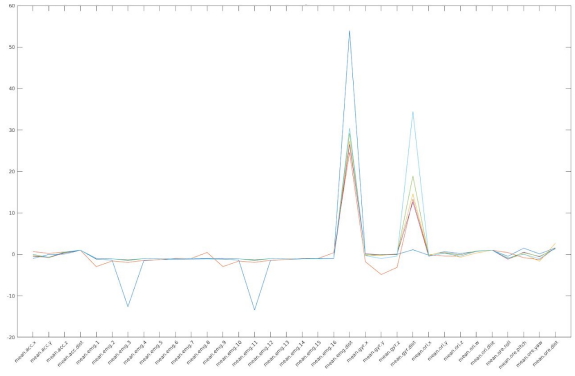
max_gyr_x	-0.52314	3	3211.42693990932
max_gyr_z	0.54705	3	3211.42693990932
max_gyr_dist	0.6129	1	145349.204837136

From the observation, we can see that our initial intuition was partly confirmed as we can observe from the figures that each action appears more or less separated.

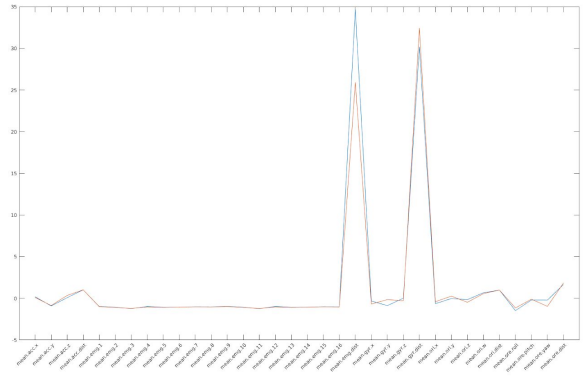
3. Average

Average is calculated by taking the sum of the values in the dataset and dividing it by the number of values in the dataset. We can use this feature as it tends to give us the central tendencies of the data.

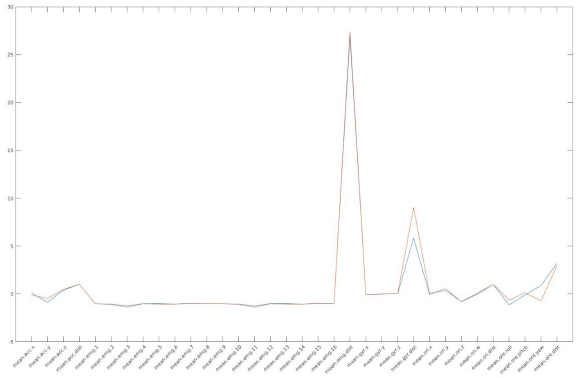
Eating AVG



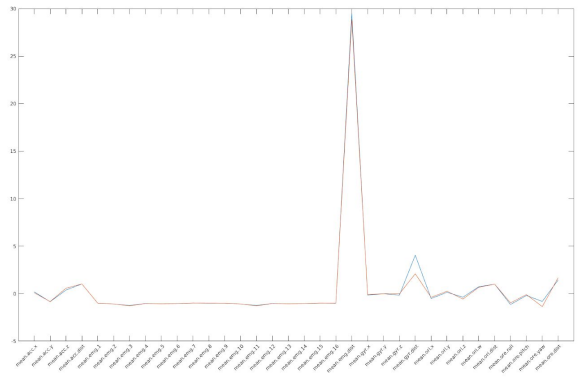
Cooking AVG



Keyboard AVG



No movement AVG



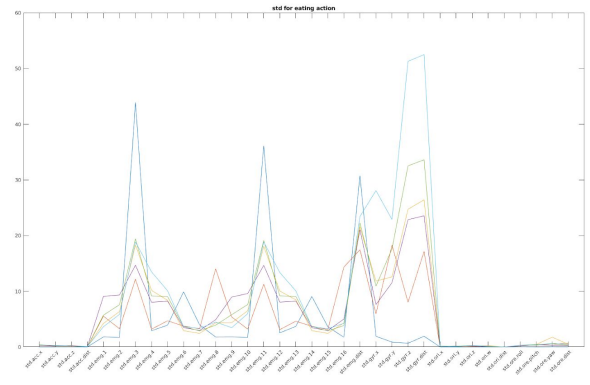
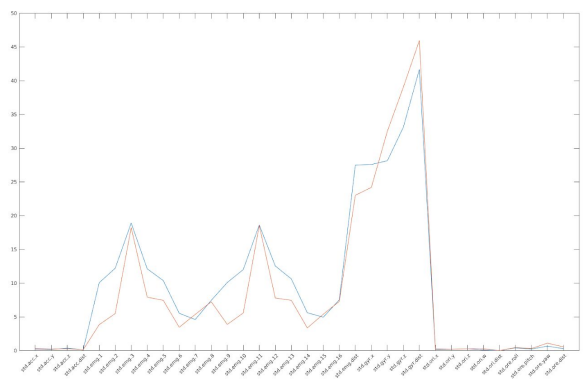
We were then able to extract the eigenvalues from the data:

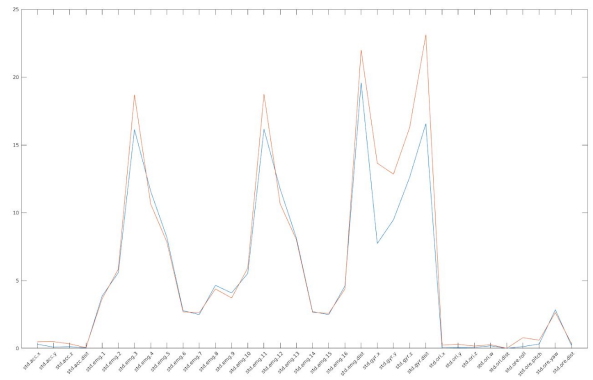
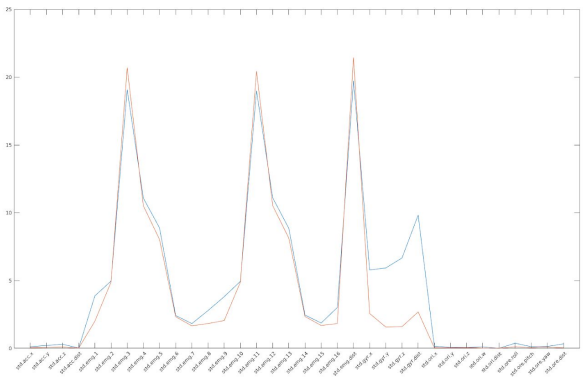
AVG Feature	Coefficient	Column	Eigenvalue
mean_emg_1	-0.0082642	5	1922.4284847171
mean_emg_8	0.0066234	5	1922.4284847171
mean_gyr_y	0.0084031	6	1529.83043728323
mean_gyr_x	-0.0072201	5	1922.4284847171
mean_ori_y	0.0032886	6	1529.83043728323
mean_gyr_z	-0.013034	5	1922.4284847171
mean_gyr_dist	0.6129	6	1529.83043728323
mean_ore_pitch	0.0060118	6	1529.83043728323
mean_ore_yaw	0.01284	6	1529.83043728323

From the observation, we can see that our initial intuition was partly confirmed as we can observe from the figures that each action appears more or less separated.

4. Standard Deviation

Standard deviation can give us the degree of variation in the data set. Standard deviation is calculated by first calculating the mean of all the numbers. Then for each number, we subtract the mean and square the result. Then we take the square root of the mean of the squared differences. While some action may contain more diverse movement (large variation, such as playing basketball) and others may have more consistent movement(small variation, such as running the marathon), STD can be useful in distinguishing these two.

Eating STD	Cooking STD
	

Keyboard STD	No Movement STD
	

We were then able to extract the eigenvalues from the data:

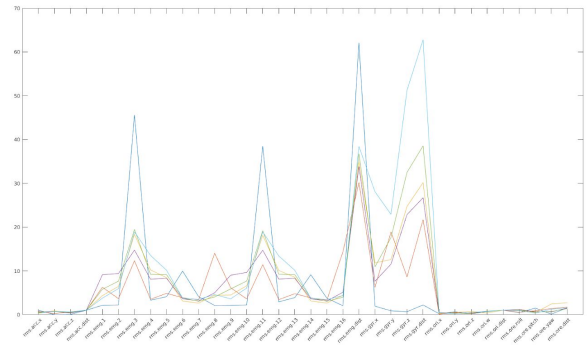
STD Feature	**Coefficient**	**Column**	**Eigenvalue**
std_emg_3	0.12264	6	1529.83043728323
std_emg_8	0.050276	5	1922.4284847171
std_emg_11	0.093633	6	1529.83043728323
std_emg_16	0.052454	5	1922.4284847171
std_gyr_z	-0.17196	6	1529.83043728323

From the observation, we can see that our initial intuition was partly confirmed as we can observe from the figures that each action appears more or less separated.

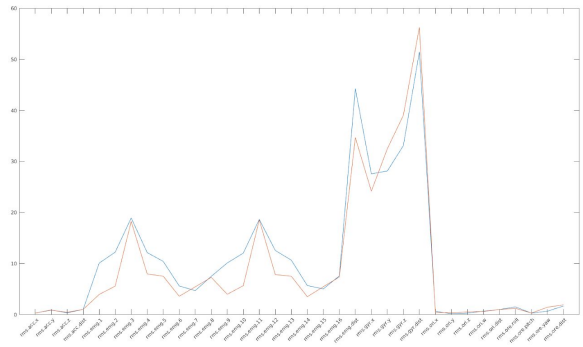
5. Root Mean Square

RMS is calculated by taking the square root of the arithmetic mean of the squares of a set of numbers. We chose Root Mean Square (RMS) as a feature extraction method for its modeling as an amplitude modulated Gaussian random process.

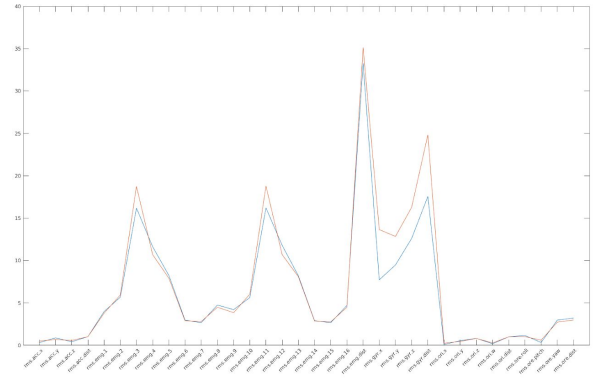
Eating RMS



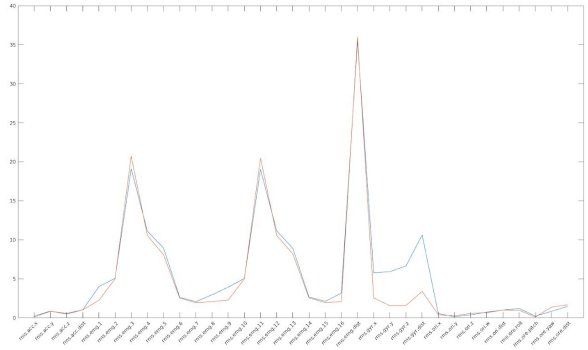
Cooking RMS



Keyboard RMS



No Movement RMS



We were then able to extract the eigenvalues from the data:

RMS Feature	**Coefficient**	**Column**	**Eigenvalue**
rms_emg_3	0.12875	6	1529.83043728323
rms_emg_8	0.049203	5	1922.4284847171
rms_emg_11	0.1022	6	1529.83043728323
rms_emg_16	0.051397	5	1922.4284847171
rms_gyr_z	-0.17266	6	1529.83043728323

From the observation, we can see that our initial intuition was partly confirmed as we can observe from the figures that each action appears more or less separated.

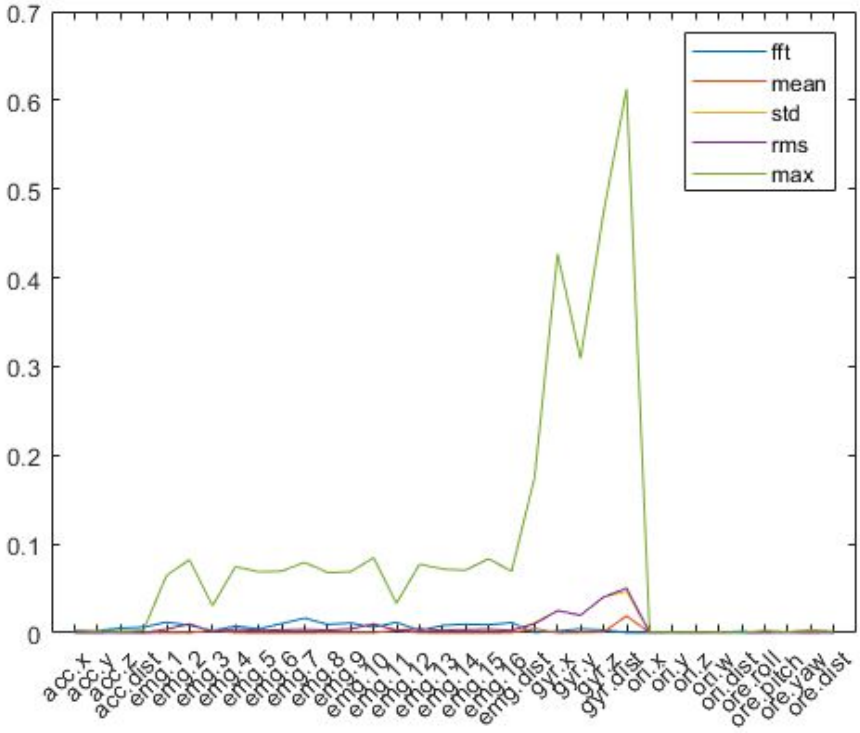
Phase Three: Feature Selection

Arranging Feature Matrix

The feature matrix is arranged as followed: The row represents a single action and the column represents features. Each row contains all the features being extracted by FFT, RMS, Average, STD, MAX.

PCA Execution

PCA results in a set of vectors that maximize variance, descending. PCA is uses this vector transformation to reduce the dimensionality of a feature space. By extracting features with a high variance, useful information in the dataset is retained while extraneous information can be safely deleted. Which is to say, information that does not have a significant impact when transformed along its vector is also information that provides for poor classification. In our MATLAB code, PCA takes in a matrix of features from different events and returns a coefficient matrix with values sorted from highest eigenvalue to lowest, a matrix of coefficient scores, and the vector of eigenvalues of the covariance matrix of the features. The larger the eigenvalue returned, the larger the variance of its corresponding eigenvector, indicating that feature is more significant than a feature that has a smaller eigenvalue.



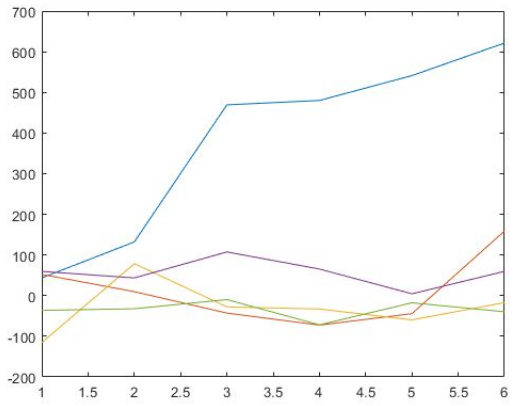
PCA Eigenvectors

A high eigenvalue signifies a high variance in the data, and that we should use the corresponding eigenvector. In MATLAB, PCA eigenvectors are displayed in order of most to least significant from left to right. If an eigenvector is completely insignificant, the MATLAB code removes it. The benefit of eigenvectors is that they allow mapping data to a more intuitive representation that can be used to reduce the effects of the Curse of Dimensionality by potentially reducing the number of dimensions we have to consider. For example, if an eigenvalue is 0 or very small compared to other eigenvalues, we can eliminate that value's eigenvector with confidence, then map our data onto the new smaller dimensionality of the remaining eigenvectors and thus improve storage and analysis performance later on.

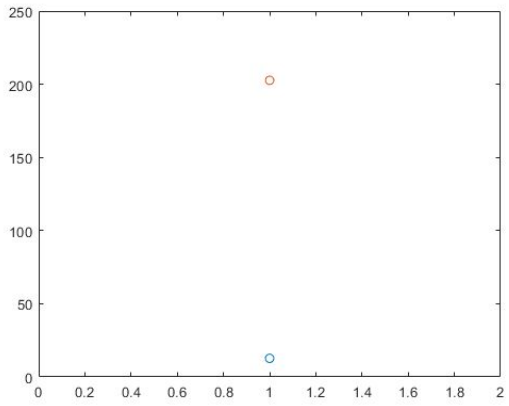
PCA Results

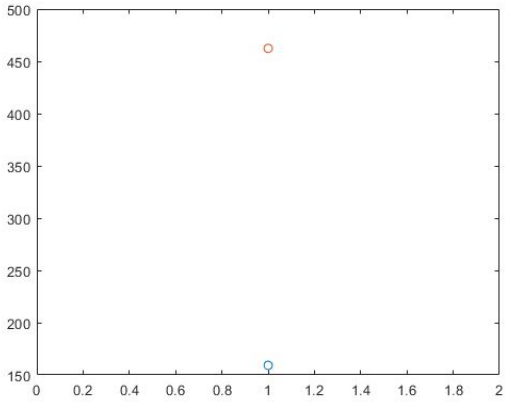
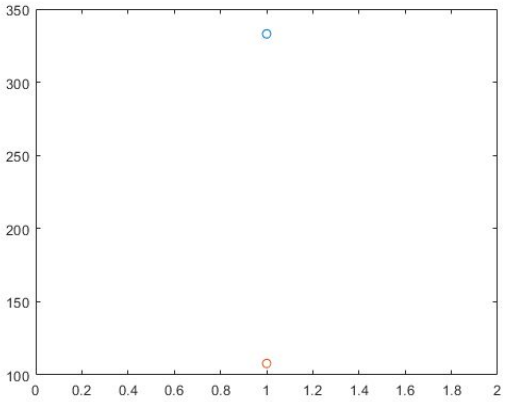
We first show the new feature matrix transformed by the eigenvectors under different activities. Now Actions in the same activity are plotted as largely separated.

Eating



Cooking



Keyboard	No movement
	

#	Eigenvalue
1	145349.204837136
2	8884.50659340742
3	3211.42693990932
4	2203.91909655853
5	1922.42848471710
6	1529.83043728323
7	949.493795581944
8	866.080810486270
9	554.612358959138
10	453.309398045812
11	172.797632157719
12	148.840312102660
13	70.3510939600754

The Coefficient Matrix is [170 x 14] and too large to be displayed here.

Score Matrix:

-405.4	-157.1	16.1	41.5	-53.2	75.1	12.8	-4.3	2.1	-26.8	-7.7	-1.4	0.2
-324.6	-111.9	0.1	49.6	109.9	-23.7	8.6	-3.1	30.0	8.5	6.8	-2.3	-0.4
-42.8	66.3	54.4	6.5	6.9	-24.4	5.9	42.5	-5.6	-31.1	7.9	25.1	9.6
-45.4	103.0	50.4	-4.8	6.6	-18.1	5.2	1.9	14.7	-7.7	-26.4	2.6	-19.1
24.9	88.0	46.0	-27.1	-23.1	-19.4	4.0	-39.1	20.4	-28.6	20.6	-20.6	1.2
149.3	-16.5	-44.6	48.1	-66.8	-65.0	25.0	33.7	13.7	15.6	-9.4	-12.4	7.2
68.5	106.3	-69.6	35.4	-27.2	30.1	-28.5	-39.1	32.8	18.1	2.4	18.4	2.9
153.8	6.2	-113.4	17.7	29.7	-24.8	10.2	-30.2	-46.7	-27.2	-3.9	0.2	-2.4
-404.4	-102.3	-8.9	-66.2	-19.8	-40.2	-77.5	4.0	-6.5	3.1	-2.8	0.1	0.2
25.3	111.5	36.1	67.5	19.0	42.2	-40.9	29.7	-31.5	16.1	5.6	-15.0	-0.7
226.8	26.5	-79.3	-88.3	30.9	55.7	12.8	45.5	19.2	-4.1	-0.5	-7.1	1.0
-163.9	33.6	58.0	-47.2	22.5	11.3	28.0	-35.0	-18.6	24.6	-17.7	-1.2	15.3
-355.8	-15.7	-1.5	-28.2	-34.1	-2.3	42.3	4.0	-20.9	31.5	22.0	8.3	-12.0
1093.5	-137.8	56.1	-4.5	-1.3	3.5	-7.8	-10.4	-3.0	7.8	3.0	5.2	-3.0

For our cut-off, we considered everything with an eigenvalue less than the sixth place to be an insignificant feature. These became our new feature set, detailed previously.

PCA Discussion

Performing PCA was very helpful in reducing the dimensionality (and thus complexity) of our feature space. We went from potentially having 170 features to only having 28, a significant improvement. Any machine learning algorithms we run on our new feature set should be much faster and potentially have fewer errors than if they were run on the full dataset. PCA can also help reveal relationships that were previously imperceptible in a dataset as large as ours.

While PCA does not eliminate noise, it does reduce it. Since noise affects all data, taking the top eigenvalues will still cause noisy aberrations in lower eigenvectors to be eliminated from the feature space. Likewise, since noise can be both low and high variance, noise that has a large variance will stand out in the PCA analysis, and will not be a good feature. However, since the purpose of PCA is not noise reduction but dimensionality reduction, using it here has achieved our goals.