SpringBoard Capstone Project 2: Human Activity Recognition
Milestone Report 1

Student: Sofy Weisenberg
Date: 03/31/20

This Capstone Project milestone report will cover the following project sub-topics:

Background

Human activity recognition (HAR) has become an in-demand capability for several growing applications over recent years. The reduced size of accelerometers and gyroscope components have made them standard in most smartphone and wearable computing devices. Additionally, sensor data collection has become less expensive (higher framerates, more channels) and data storage options have grown (use of cloud-based storage, etc). Sensor-based activity recognition researchers believe that by empowering ubiquitous computers and sensors to monitor the behavior of agents (under consent), these computers will be better suited to act on our behalf. For example, these may include monitoring of activities of those suffering from chronic diseases (such as diabetes or COPD) or monitoring sleep and exercise patterns.

Data Set

- Data was gathered for 30 subjects, aged 19-48 years old. Each subject performed 6 activities while wearing a smartphone (Samsung Galaxy S2) on their waist: laying, sitting, standing, walking, walking upstairs, and walking downstairs.
- The smartphone contains an accelerometer and a gyroscope for measuring 3-axial linear acceleration and angular velocity respectively at a constant rate of 50 Hz, which is sufficient for capturing human body motion.
- The data was manually labeled using corresponding video recordings of the experiments.

Data Preprocessing/Feature Engineering

The dataset available through Kaggle has already been processed and organized into a format usable by various machine learning techniques. Specifically the following steps were performed:

- The sensor signals (accelerometer and gyroscope) were pre-processed by applying noise filters and then sampled in fixed-width sliding windows of 2.56 sec and 50% overlap (128 readings/window).
- The sensor acceleration signal, which has gravitational and body motion components, was separated using a Butterworth low-pass filter into body acceleration and gravity. The gravitational force is assumed to have only low frequency components, therefore a filter with 0.3 Hz cutoff frequency was used.
- From each window, a vector of features was obtained by calculating variables from the time and frequency domain.

This resulted in a dataset with the following attributes provided for each record:

- Triaxial acceleration from the accelerometer (total acceleration) and the estimated body acceleration.
- Triaxial Angular velocity from the gyroscope.
- A 561-feature vector with time and frequency domain variables.
- Its activity label.
- An identifier of the subject who carried out the experiment.

Additional feature engineering details are provided for this dataset on the UCI website (features_info.txt from the downloadable data):

The features selected for this database come from the accelerometer and gyroscope 3-axial raw signals tAcc-XYZ and tGyro-XYZ. These time domain signals (prefix 't' to denote time) were captured at a constant rate of 50 Hz. Then they were filtered using a median filter and a 3rd order low pass Butterworth filter with a corner frequency of 20 Hz to remove noise. Similarly, the acceleration signal was then separated into body and gravity acceleration signals (tBodyAcc-XYZ and tGravityAcc-XYZ) using another low pass Butterworth filter with a corner frequency of 0.3 Hz.

Subsequently, the body linear acceleration and angular velocity were derived in time to obtain Jerk signals (tBodyAccJerk-XYZ and tBodyGyroJerk-XYZ). Also the magnitude of these three-dimensional signals were calculated using the Euclidean norm (tBodyAccMag, tGravityAccMag, tBodyAccJerkMag, tBodyGyroMag, tBodyGyroJerkMag).

Finally a Fast Fourier Transform (FFT) was applied to some of these signals producing fBodyAcc-XYZ, fBodyAccJerk-XYZ, fBodyGyro-XYZ, fBodyAccJerkMag, fBodyGyroMag, fBodyGyroJerkMag. (Note the 'f' to indicate frequency domain signals).

These signals were used to estimate variables of the feature vector for each pattern:
'-XYZ' is used to denote 3-axial signals in the X, Y and Z directions.

- tBodyAcc-XYZ
- tGravityAcc-XYZ
- tBodyAccJerk-XYZ
- tBodyGyro-XYZ
- tBodyGyroJerk-XYZ
- tBodyAccMag
- tGravityAccMag
- tBodyAccJerkMag
- tBodyGyroMag
- tBodyGyroJerkMag
- fBodyAcc-XYZ
- fBodyAccJerk-XYZ
- fBodyGyro-XYZ
- fBodyAccMag
- fBodyAccJerkMag
- fBodyGyroMag
- fBodyGyroJerkMag

The set of variables that were estimated from these signals are:

- mean(): Mean value
- std(): Standard deviation
- mad(): Median absolute deviation
- max(): Largest value in array
- min(): Smallest value in array
- sma(): Signal magnitude area
- energy(): Energy measure. Sum of the squares divided by the number of values.
- iqr(): Interquartile range
- entropy(): Signal entropy
- arCoeff(): Autorregresion coefficients with Burg order equal to 4
- correlation(): correlation coefficient between two signals
- maxInds(): index of the frequency component with largest magnitude
- meanFreq(): Weighted average of the frequency components to obtain a mean frequency
- skewness(): skewness of the frequency domain signal
- kurtosis(): kurtosis of the frequency domain signal
- bandsEnergy(): Energy of a frequency interval within the 64 bins of the FFT of each window.
- angle(): Angle between two vectors.

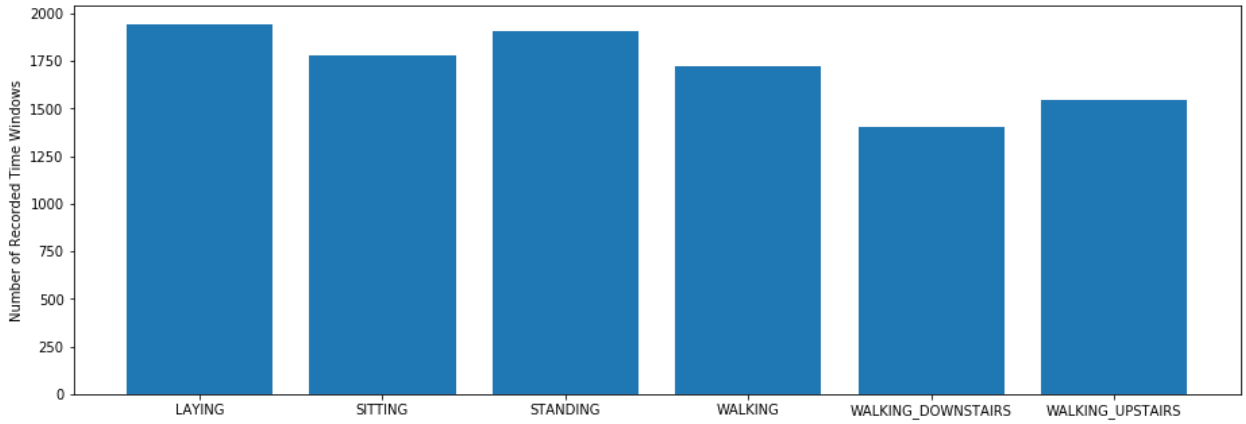
Additional vectors were obtained by averaging the signals in a signal window sample. These are used on the angle() variable:

- gravityMean
- tBodyAccMean
- tBodyAccJerkMean
- tBodyGyroMean
- tBodyGyroJerkMean

Initial Data Exploration

Loading the dataset, it can be seen that there are a total of 10299 rows and 563 columns, with no missing values. The feature data is all data type float64, and the last 2 columns are the subject number (int64) and the activity class (object).

The various activity classes are approximately evenly distributed, since each subject performed each of the six activities. This means that for classification modeling, this is a balanced (rather than an imbalanced) dataset.

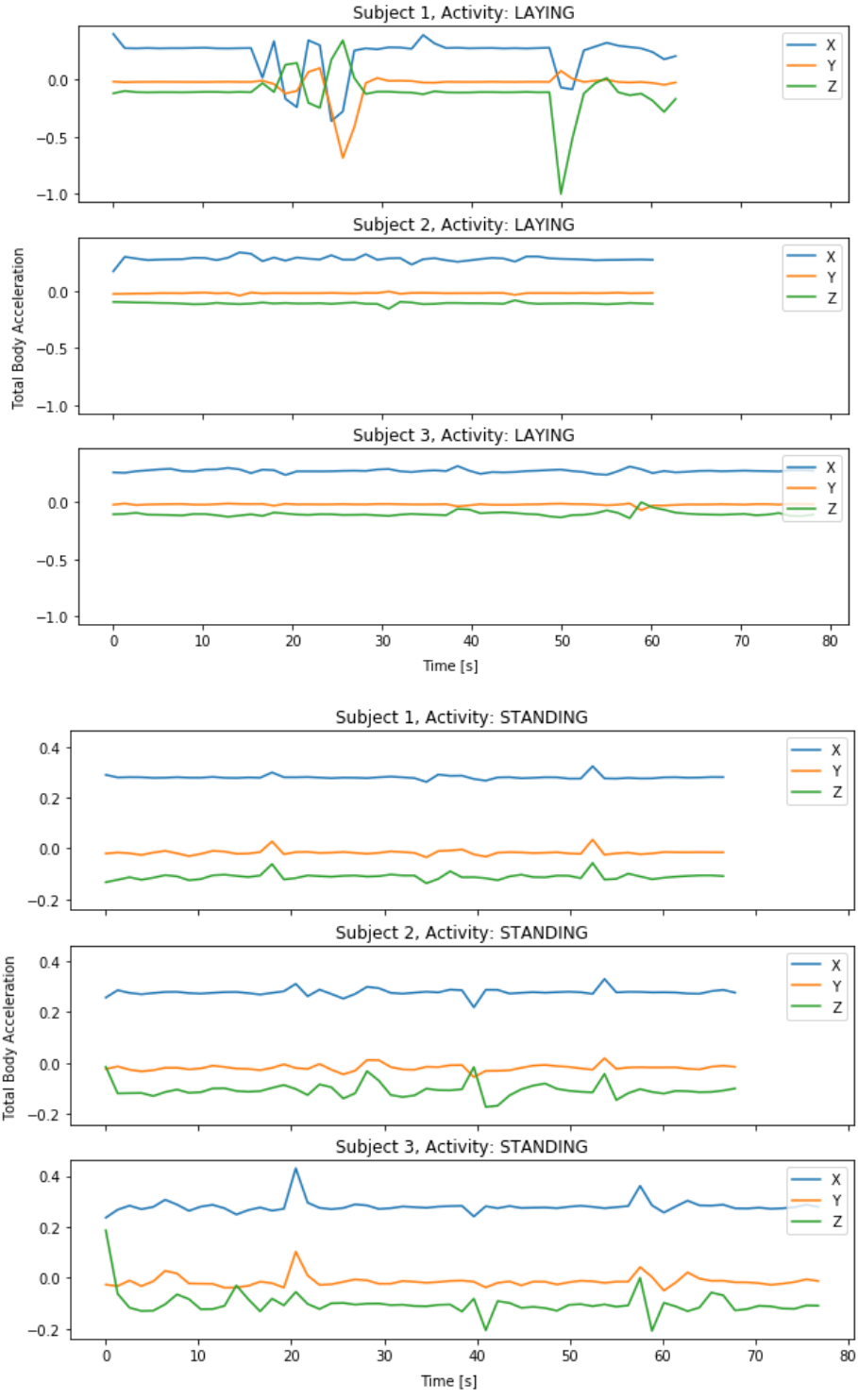


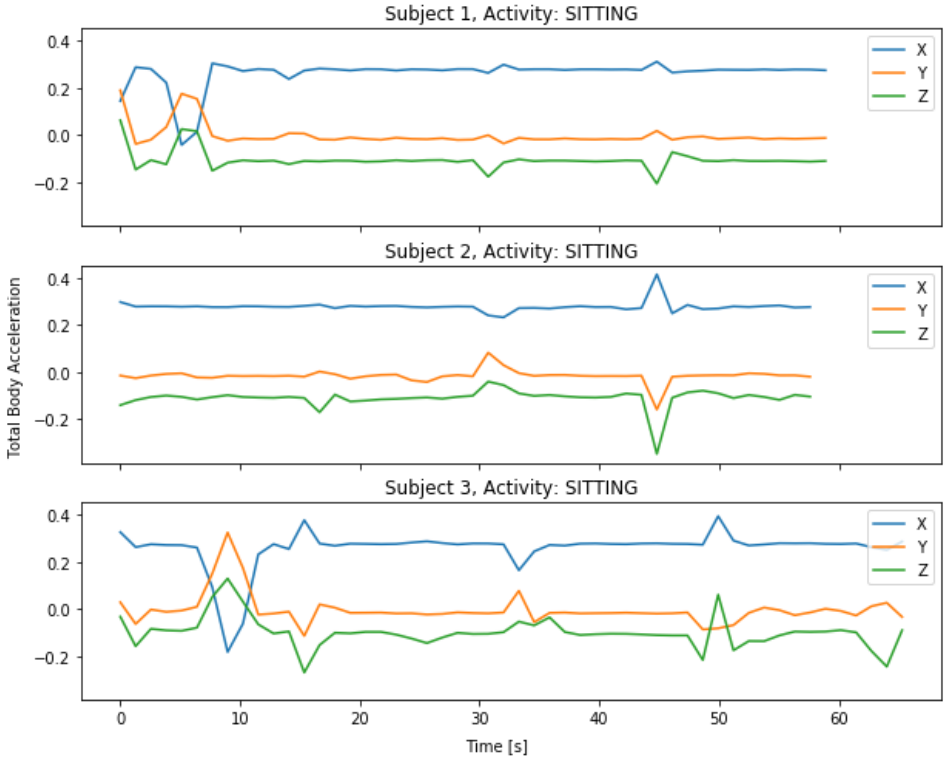
Next, it would be helpful to see a summary of the descriptive statistics for the data. This is done using the .describe() method. From the results of the .describe() method, it appears that the data has already been scaled to a range of [-1, 1] for all feature columns which will be helpful for both data visualization and machine learning. Additional scaling, for example standardization in which the mean is zero and standard deviation is 1, may be additionally performed during the modeling phase of the project.

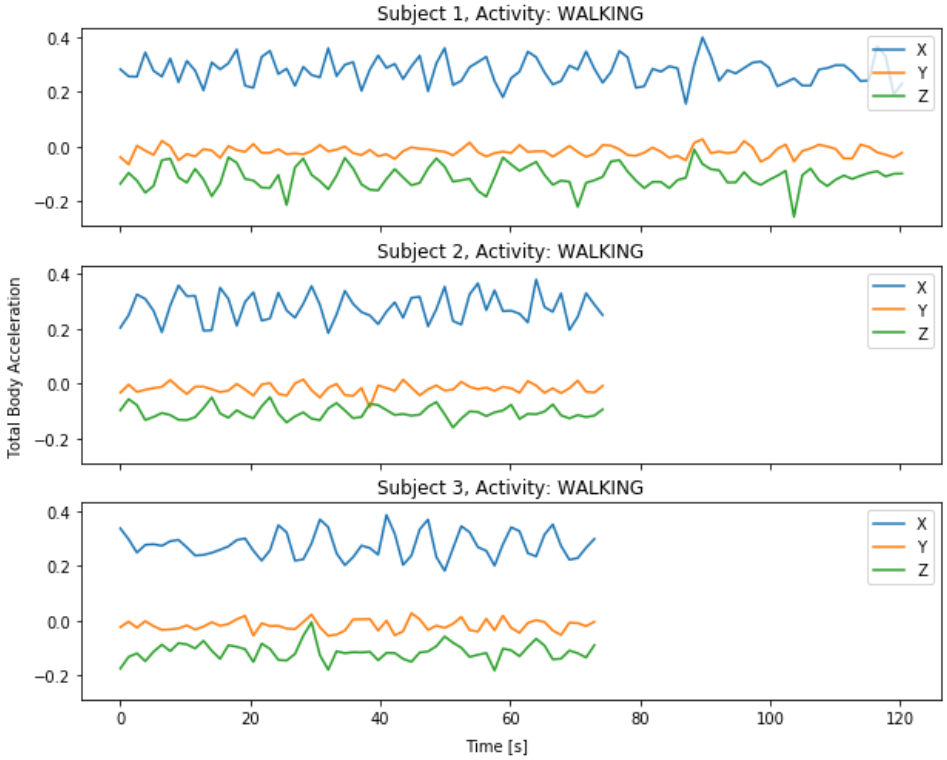
Plotting Example Sensor Data

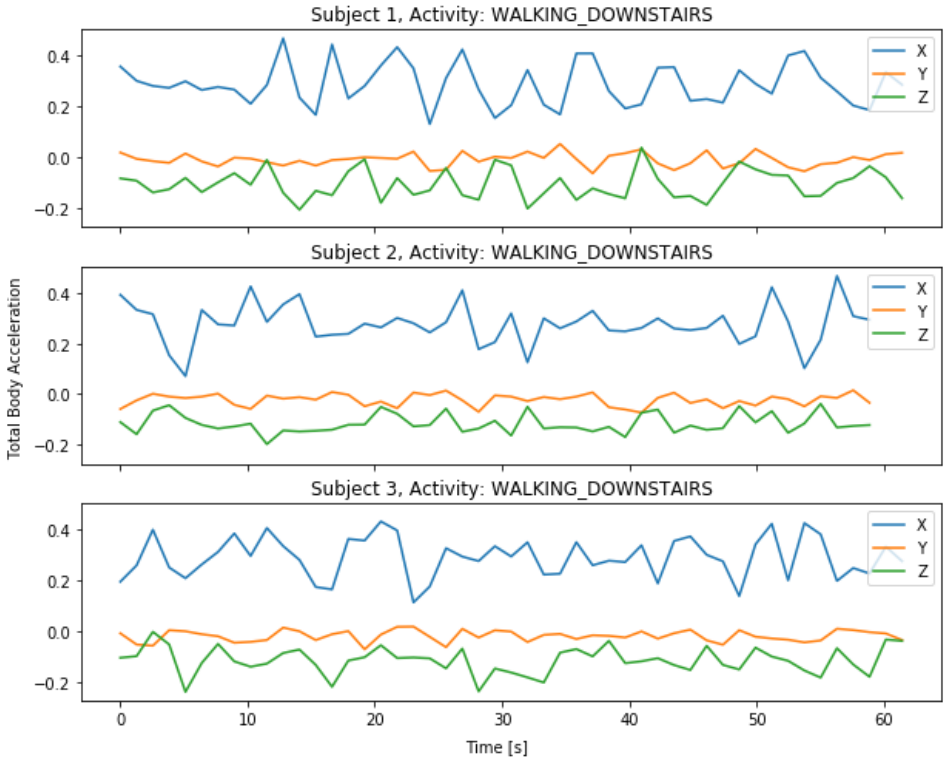
To get a feeling for the type of data, it would be helpful to plot a few examples of the various activities across several subjects. Total mean acceleration in the x,y,z directions is a good starting point.

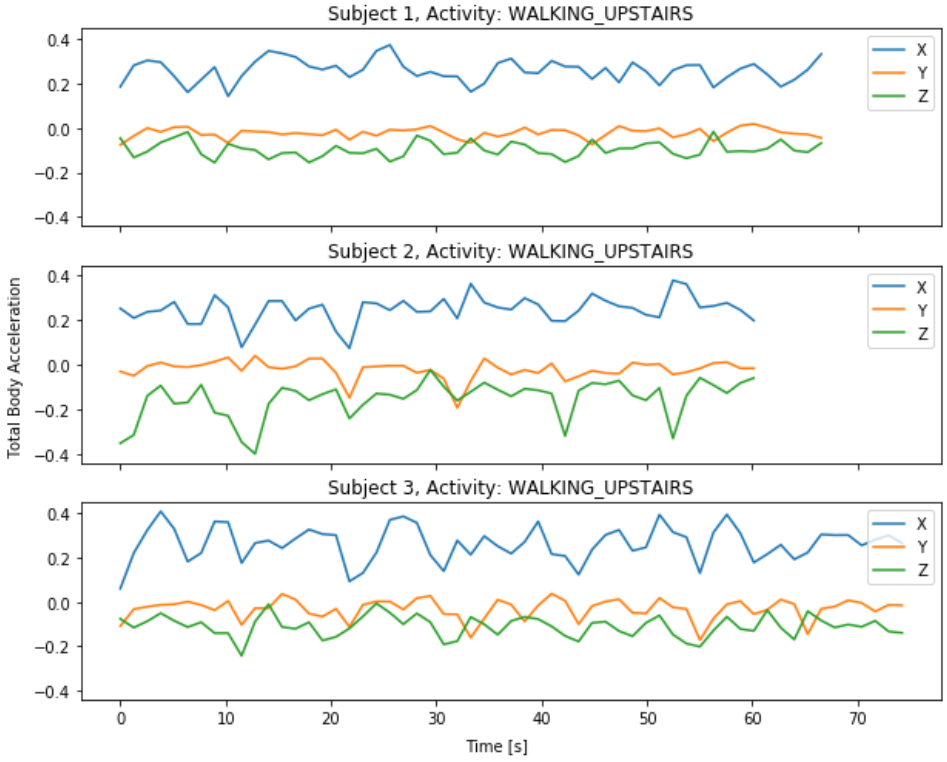
The following plots explore this feature for each of the 6 activities and each of the first 3 subjects.




Subject 1, Activity: SITTING
Subject 2, Activity: SITTING
Subject 3, Activity: SITTING
Total Body Acceleration
X
Y
Z
0.4
0.2
0.0
−0.2
0
10
20
30
40
50
60
Time [s]


Subject 1, Activity: WALKING
Subject 2, Activity: WALKING
Subject 3, Activity: WALKING
Total Body Acceleration
X
Y
Z
0.4
0.2
0.0
−0.2
0
20
40
60
80
100
120
Time [s]


Subject 1, Activity: WALKING_DOWNSTAIRS
Subject 2, Activity: WALKING_DOWNSTAIRS
Subject 3, Activity: WALKING_DOWNSTAIRS
Total Body Acceleration
Time [s]
X
Y
Z


Subject 1, Activity: WALKING_UPSTAIRS
Subject 2, Activity: WALKING_UPSTAIRS
Subject 3, Activity: WALKING_UPSTAIRS
Total Body Acceleration
Time [s]
X
Y
Z

Some observations from the plots above:

- Each subject recorded the same activity for a different period of time.
- The laying activity, as expected, has the lowest signals (mostly flat). Except subject 1 who had an erratic (unusual) signal.
- Standing and sitting have signals of similar magnitude, which might be a source of classification error.
- Walking upstairs seems to produce more regular cycles of movement than walking downstairs.

Visualizing Separability of Classes

From the single attribute type plotted above (total body acceleration), some differences between the classes are immediately noticeable. However, it does not give a clear picture of the overall dataset and how the rest of the attributes contribute to the separability of the activity classes.

Two visualization techniques, using different dimensionality reduction methods, will be implemented and compared to get a qualitative indication of the class separability using the full dataset. The techniques are as follows (source):

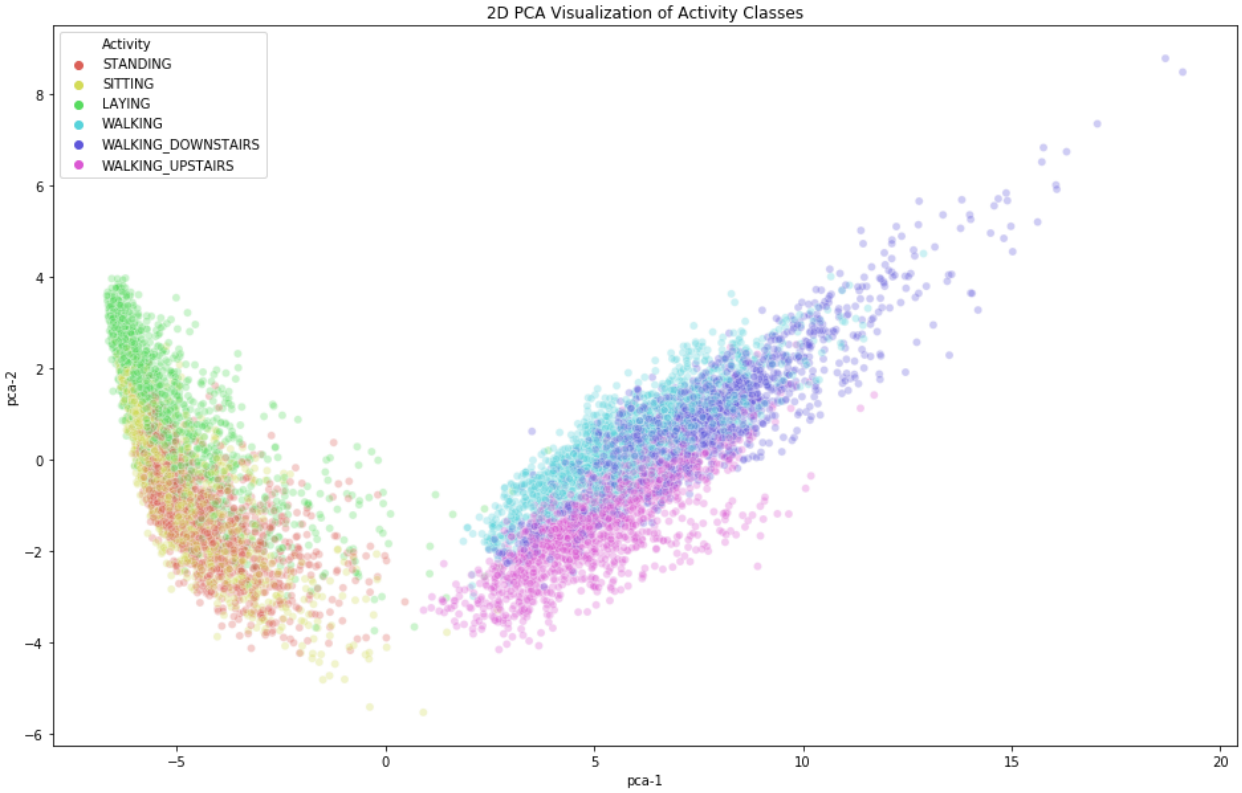
- Principle component analysis (PCA) uses the eigenvectors of the covariance matrix to reduce the dimensionality of the data. The eigenvectors point in the directions of maximum variation in a dataset, preserving as much of the variability in the data in as few dimensions as possible. Two or three of the principle components may be plotted for visualization purposes.
- t-Distributed Stochastic Neighbor Embedding (t-SNE) is a probablistic technique for dimensionality reduction, which minimizes the divergence between two distributions: a distribution that measures pairwise similarities of the input objects and a distribution that measures pairwise similarities of the corresponding low-dimensional points in the embedding.

t-SNE is computationally heavy for very high dimensional data and will likely require an initial dimensionality reduction before execution, for example PCA to the first 50 principle components.

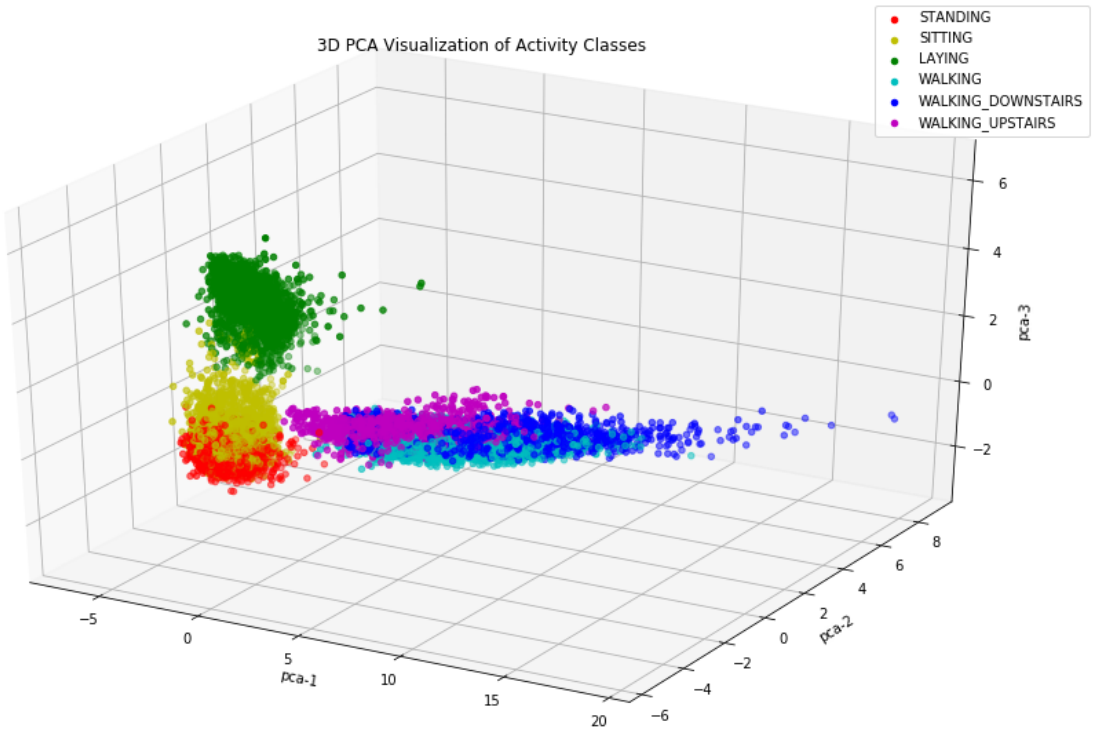
PCA

The PCA analysis was quite fast (~0.20 s), requiring relatively low computational resources to execute. The first 3 principle components represent nearly 71% of the overall variation in the dataset. This is quite high and should hopefully allow for producing some useful visualizations.

The following figures are two rotations of the 2D PCA plots (with 1st/2nd and 1st/3rd principle components) and the 3D plot of all 3 principle components):


2D PCA Visualization of Activity Classes
Activity
STANDING
SITTING
LAYING
WALKING
WALKING_DOWNSTAIRS
WALKING_UPSTAIRS
pca-2
pca-1
8
6
4
2
0
−2
−4
−6
−5
0
5
10
15
20


2D PCA Visualization of Activity Classes (Rotated)
Activity
STANDING
SITTING
LAYING
WALKING
WALKING_DOWNSTAIRS
WALKING_UPSTAIRS
pca-3
pca-1
6
4
2
0
−2
−5
0
5
10
15
20



As can be seen from the above plots, the “active” classes (walking, walking upstairs, walking downstairs) are very separable from the “non-active” classes (laying, standing, sitting).

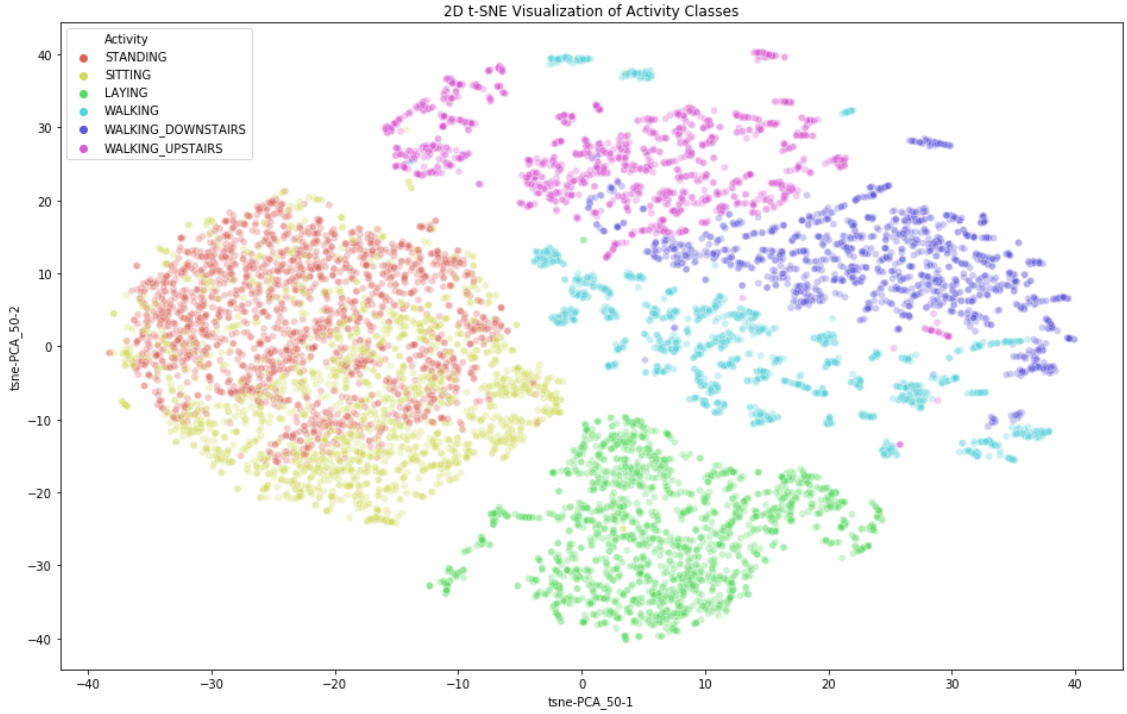
- Within the non-active group, the laying class is much more separated than the sitting and standing classes.
- Within the active group, the classes are less separated, with the walking class bridging the other two.

tSNE

The t-SNE approach to class separability visualization took nearly 600x the runtime of PCA. Several combinations of perplexity and n_iterations were tried with no significant difference in the overall visualization appearance or KL divergence. The algorithm appears to have converged and produced some plausible clusterings, which offer a bit more visual separation than PCA, though standing and sitting classes still appear heavily intermixed. However, the drawback of t-SNE is in its interpretability: since Euclidean distance between points and between clusters is not necessarily indicative of relative correlation strengths. Overall, it seems PCA could be a sufficient technique for visualization in this case.

The following visualization was produced with this method:

Additionally, it is possible to combine the two techniques (PCA with first 50 principle components and t-SNE) and examine the result. The cumulative explained variation for 50 principal components is 92%. The following is the resulting plot:



When combining the two techniques, the runtime of t-SNE is reduced by about 68%. The visualizations are very comparable, with some expected rotation. There does not appear to be any significant loss of relational/probablistic robustness.

This would indicate that there is a computational advantage to running PCA first and then t-SNE. However, with PCA offering better interpretability than t-SNE, as well as being faster and deterministic, it seems to be the best option for this analysis.