Dante Smith, PhD

You can find my work for this process at the following GitHub repository. Thank you for the opportunity to apply and the fascinating problem statement!

https://github.com/djsmith17/pison-tech-assess

The data provided are time-series data recorded from a Pison device during five different types of activities. There are two channels of EMG electrodes as well as a gyroscope, accelerometer, and a quaternion. We are also provided with a body position label, and a repetition number, which I am interpreting as a trial number. It is requested that I build a classifier to identify different classes of wrist movement based on the read outs of these sensors. Based on inspection of the data there appears to be three different repetitions (trials) of each body movement. While it is not defined, I believe it is safe to say there is only one specific wrist movement per body movement trial. Therefore, from the time-series data there are 15 wrist movements to identify, and at most 15 different classes of movement. I think it is likely there will be less wrist movement classes to classify.

It will be somewhat difficult to train a classifier that is not over-fit to this training data without more sample repetitions, but I think for now it safe to identify how many different classes of wrist movements are included here based on the differences between the recording sets. I will be considering each repetition as its own sample to consider, and therefore will need to perform some feature-engineering to identify shapes, patterns, and relations of the time-series sensor data that disambiguate themselves between repetitions. There are 15 individual wrist movements to identify but 14 different time-series for each wrist movement to pull out features. My main goal will be to directly compare the types of waveforms seen in each of these sensor sources and have the comparisons be time-invariant.

Data science has attempted to solve this problem with an algorithm called Dynamic Time Warping (DTW) which compares two time-series shapes and computes a distance between them that describes how alike the shapes while remaining time-invariant and even compression/stretch invariant. I will be using DTW to first assess the relative similarities (distance measures) between each of the 15 wrist movements within 10 of the 14 sensor features. Specifically, I will be focusing on all but the quaternion features. I have decided to do this because to the best of my knowledge, the quaternion projections represent similar data to that of the accelerometer and gyroscope. Based on the outcomes of this process I will be left with 10 confusion matrices (15 x 15) that describe the distances how each body movement/repetition time-series relates to each sensor reading collected.

From each of these confusion matrices, I will need to perform a form of decomposition that applies a normalization of the distances (differences) across repetition/body movement. Essentially, for each of the 10 confusion matrices, I need to decompose the 15x 15 matrix into a 15 x1 description of how ‘different’ or ‘unique’ a given repetition/body movement combo is compared each other within a sensor recording type. I chose to simply to take the mean across the rows of the confusion matrix to perform this decomposition.

From this decomposition we are left with 10 sensor vectors of length 15 that roughly describe the ways in which one time-series recording of a sensor type differ from the others time-series recordings. Put together, this 15x 10 matrix is in perfect shape to perform machine learning on. At this stage its advantageous to reduce the complexity of our data and we can

reduce the number of variables we have using Principal Component Analysis (PCA). While we only have 15 observations (repetition/ body position combo) to select from, I was still able to reduce the number of variables down to 3 principal components.

From here, we are ready to start clustering. I think a simple k-means clustering algorithm should be sufficient for now. K-means is a simple clustering algorithm that identifies the centroid of the distribution of values from the features included in the data set. It's difficult to tell exactly how many clusters of data points satisfy the data, so in fact its best to try several cluster counts and observe the Within-Cluster Sum Squares (WCSS) to identify the relative error of a cluster's centroid to the true centroid of the clustered data points. This exercise produces an elbow curve graph which identifies the point at which the number of clusters fails to further reduce the level of error. From my own WCSS plot I found there to roughly 6 distinct clusters of data points. From my processes, I have therefore determined there to be 6 classes of wrist movements in this data set.

From my data science process, I have deduced there to be 6 classes of wrist movements. In my scripts I have sampled one example from each class and created a gif of sensor movement in that class to better understand the changes occurring. By viewing the gifs, and through my own understanding of wrist mechanics, I have given a guess to the description of this movement class. The recordings in this data set belong to the following classes:

Table 1: BL refers to Body Movement label and rep refers to repetition number

Class	**Dataset Sample**	**Description**
Class 0	BL0, Rep1; BL1, Rep1; BL2, Rep1; BL3, Rep1	Stirring Motion in X/Y Plane
Class 1	BL0, Rep3; BL1, Rep3; BL2, Rep3; BL3, Rep3	Cross Body Wave
Class 2	BL4, Rep2	Figure-8 trace in Y/Z Plane
Class 3	BL0, Rep2; BL1, Rep2; BL2, Rep2; BL3, Rep2	Figure-8 trace in Y/Z Plane
Class 4	BL4, Rep3	Cross Body Wave
Class 5	BL4, Rep1	Stirring Motion in X/Y Plane

Final Thoughts:

As mentioned previously, there may be too few samples here to make this fully translatable. It is also difficult to ascertain the performance of this model without knowing the true labels of the wrist motions. I approximated the performance of the K-means using the WCSS error, and I believe I minimized the error to the point that the model is performing as well as it could be.

I have generated subplot figures of the features used to generate my model, as well as 3D animations of the accelerometer and gyroscope sensors. You can find those files in the folders of the repo titled *FeaturePlots* and *WristGifs* respectively. From visual inspection of the signals, it may stand to reason that there are only 3 classes of wrist movement instead of 6. It is certainly true that the wrist movements seem to cluster around the repetition number, with the repetitions of Body Movement 4 (Running) seemingly singled out. I could imagine a scenario where the running action created signal that was different enough from the other classes to generate their own cluster.