Machine Learning

**Overview**

In order to classify the input sensor data as a specific gesture, we decided to train a multiclass classification model to recognize 4 gestures: Flat Downward Palm (label 0), Flat Left Palm (label 1), Tight Fist (label 2), and Peace Sign (label 3).

**Dataset**

In order to collect and label the data, we implemented a python command-line program to prompt user to hold a gesture for a continuous period of time while collecting the sensor data sent from the Arduino during this period of time. The user would discontinue sensor reading and label the data passed so far by toggling a switch attached to the Arduino microcontroller.

For our demonstration, we used a dataset with 417 rows of labeled data. The distribution of different gestures within the dataset is approximately evenly distributed. We have tested our model with much more data (3,000 rows) collected in such setting, but the performance remained the same.

Below are several sample rows of our data:



**Data Preprocessing**

Because the pressure sensors generate a large amount of noise beyond our control, we performed data preprocessing for the data to be more easily workable.

First, when the data is read by Arduino from the sensors, we only pass the average of a size-10 sliding window. This has reduced the noise significantly.

Second, before we feed the data to either train the model or predict gesture using an existing model, we perform feature normalization to the data. The feature normalization has significantly increased the out-of-sample performance of our trained model.

Third, we split the dataset into 80% training and 20% testing in order to evaluate the performance of our model.

**Model Selection**

Following the philosophy of Occam’s razor, we decided to use Support Vector Machine (SVM) as our training model. We selected the linear kernel or our SVM. For the multiclass classification, we used Scikit-Learn’s built-in “one verses rest” strategy to reduce the overall training time. After comparing the performance of linear, polynomial, and RBF kernels on the test data as well as testing on real-time gesture input form the gesture band, we discovered that linear model achieved the best performance on the given dataset both in terms of test data accuracy as well as real-time gesture data accuracy.



**Performance and Challenges**

Our model is able to predict a user’s hand gesture with accuracy around 87% during real testing with sensor, despite taking off the gesture band and reattach the band after training. However, after some testing, we realize that the robustness of our model can be improved. A model trained on a single person’s arm is not able to generalize to predicting gesture for another user. Besides, if the user is in sitting down while labeling the data, the model is not able to perform well when the user is standing up.

**Potential Improvements**

In order to increase the robustness of our machine learning model, we can collect more user data in separate settings perhaps by specifying different body positions for the user to be in while streaming the gesture data to the database. Besides, we can collect a larger number of data from different people and train the model to help the model better generalize to gestures performed by separate individuals.

