EMG Research Report

Ziyi Jin

Data Collection

In order to build a classifier to classify typing and non-typing movements, we have to collect the EMG, Gyroscope and Accelerometer data when a person is typing and not typing. The data were collected using Myo armband with the sampling frequency at about 160HZ and can be categorized into 5 classes, including typing(positive), non-typing(negative), drinking(negative), grabbing(negative), eating(negative). The position where the Myo armband was put on the forearm is indicated in the right figure. All these raw data were saved in CSV files. The data in each row contains 15 elements and the array representation looks like [timestamp, channel 1~8, GyroX~Z, AccelerometerX~Z, Active]. The last entry "Active" indicates whether there was a movement performed.

The code that is used to connect to the Myo band and collect the dat is stored in emg-data-sample.cpp file. This code should be build together with Myo SDK in order to work properly. The detail can be found in the Myo armband developer website.

The raw data is stored in the Data folder.

Data Visualization

The python code to visualized the data can be found in the project folder on Github. The code can be used to plot the EMG data from channel 1 to 8 on one single graph.

The visualized data for each data category can be founded in the Plot folder.

Data Filtering

The python code to filter the data can also be found in the project folder on Github. There were three most commonly used filters being implemented, including low-pass, high-pass and bandpass filter. The API to access the implementation can be found in the code documentation

Feature extraction functions

According to the research paper, I implemented 8 functions to extract the time-domain features and the code is stored in the functions.py file.

In order to create proper feature and label sets from the raw data, one should call load_data function in the loadData.py file. This function will read the dataset that is indicated by the user and return a feature vector array and a label vector that is built from the dataset. The user can also choose the window size by changing the function parameter. The default window size is 30. If the data contains X rows, the windowed feature will contain lower_bound(X / window_size) rows.

There are currently 98 features(14 raw data entry * 7 feature extraction function) in each feature vector. The feature that should be extraced by the function diff_mav() has not been added to the feature vector.

Test Prediction

I have tested the prediction result using SVM model(with half of the features used). The testing code is in svm.py file. The overall accuracy is about 65% to 75%, with the different testing and training set percentage arrangement and the dataset random selection. This result is closed to the value on the research paper.