Final Results Writeup

Sam Coleman and Rishita Sarin

Format of Data

JuniperSun has 10 different collection structs, each with 18 trials (180 trials in total). Labels (rock, paper, or scissors) are stored in field "epochlabelscat." The data is stored in "data" field in a [3x1500X18] matrix where it is [channel X value X trial]. The times are stored in the "alltimes" field which is [1500 X 18] where it is [timeStep X trial].

Preliminary Understanding of Data

To understand the given data, we took the mean of rock, paper, and scissors respectively and plotted them separately for each channel (see below). We determined that channel 3 data looks very similar for rock, paper, and scissors, therefore our algorithm may have a hard time distinguishing between them. Thus, our focus for this sprint was primarily on channels 1 and 2. (Please note that we conducted feature extraction as well as some feature selection on all channels to test our hypothesis regarding channel 3).

Pre-Processing

Besides the pre-processing already performed when collecting data, we mean centered our data across the three channels. All feature extraction was done on the pre-processed data. The mean centered data was then split into test and train data. We held out 20% of the data for the testing set.

Feature Extraction

We chose about 20 potential features to investigate. These were a mix of features we spoke about in class and features mentioned in research papers about EMG.

The features we extracted are: mean, maximum value in signal, mean of absolute value in signal, root mean square (rms), variance, average amplitude change (aac), waveform length (lc), willison amplitude (wamp), zero crossing (zc), integrated EMG (iemg), simple squared integral (ssi), number of peaks (np), skewness (skew), difference absolute standard deviation value (dasdv), kurtosis (kurt), difference absolute mean value (damv), and log detector (ld).

All feature extraction code is in extract_features.m as a class to allow easy feature extraction across files for any set of data. After extracting features, we saved the extracted features to a mat file for easy use in the future.

Feature Selection

Note: the time window for extracting features was the whole window for all features. Additionally, 5 fold cross validation was used throughout to help prevent overfitting. Link to spreadsheet holding all featutre selection documentation

Method 1 - Sam

With a basic SVM model, we tested the accuracy with all the features by themselves and included all three channels. Looking at those results, mean absolute value and root mean square had the highest overall accuracies and had pretty consistent true positive rates across rock, paper, and scissors. wethen, using the same algorithm, abs_mean and rms together and separate in multiple combinations of included channels. From this, it was clear that channel 3 had the lowest accuracy, and the inclusion of channel 3 lowered accuracy overall. Thus, we moved forward not including channel 3 in most trials. We continued to test abs_mean and rms using different algorithms.

Fitted discriminant analysis seemed like a promising algorithm, so we went through a similar process testing each feature individually (with just channels 1 and 2 included) with fitted discriminant analysis. Using fitted discriminant analysis, with default parameters, iemg, rms, ss, and variance seemed like promising features with a high overall accuracy and consistent true positive rates for all 3 moves. We tested all possible combinations of these four features (just including channels 1 and 2). Of these combinations, including rms and ss as well as all 4 produced the best results.

Method 2 - Rishita

We started with one feature (acc) and added one feature at a time. If the added feature increased validation classification accuracy, we kept it in our next run, else we removed it. Once we reached the end of our list of features with this addition method, we reversed the list and began to remove one feature at a time (starting with acc). If removing the feature decreased the validation classification accuracy, we added it again in our next run, else we removed it for the remainder of our runs. At the end of this list of features with this subtraction method, we will have a short list of features which provide the best validation classification accuracy for our data set. With our given features, we landed with two feature lists with a high validation classification accuracy of 72.92%. They are acc + absmean and acc + absmean + kurt + mean. More details about the different feature lists tested and their respective results may be found here.

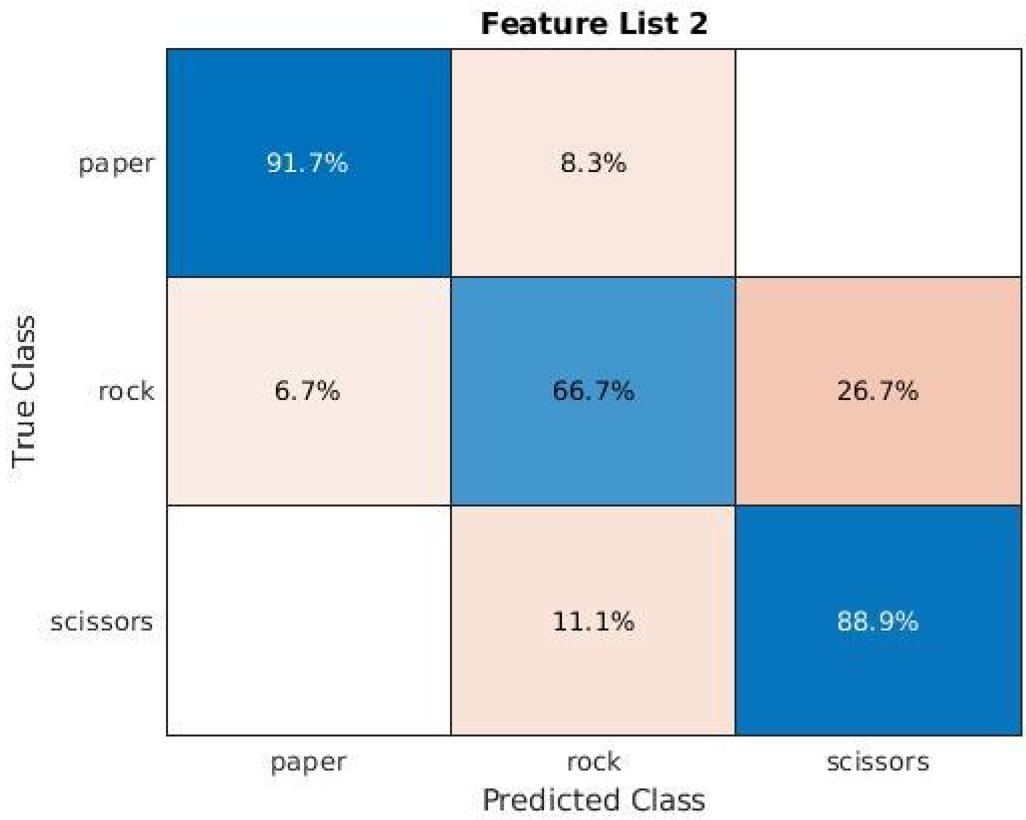
Sam determined that we would achieve higher accuracies with the exclusion of channel 3 data. Therefore, we retested our two feature lists with only channel 1 and channel 2 data. This resulted in higher accuracies of 75% and 74.31% respectively.

Final Results

After looking at the accuracies for all the algorithms for our training data, we decided to test the 3 sets of best feature lists on our test data. The first is Average Amplitude Change (AAC) and Mean of Absolute Value (abs_mean). The second is Root mean square (RMS). The third is Average Amplitude Change (AAC), Mean of Absolute Value (abs_mean), Kurtosis (kurt), and Mean. For all of the feature lists, we are only using channels 1 and 2. Additionally, we are using a trained svm classifer with no additional parameters.

Feature List 1 (AAC & abs_mean) had a testing accuracy of 86.11%.

Feature List 2 (RMS) had a testing accuracy of 80.56%.



Feature List 3 (AAC, abs_mean, kurt, mean) had a testing accuracy of 83.33%.

As the confusion matrices show, feature 1 has the highest accuracy with 86.11%.

Next Steps

1. Extract additional features after reading more research papers
2. Go through feature selection using more methods to determine. different trained classifiers we can use. Vary features included along with algorithm used.