Classifying Imaginary Movements from an 8x8 ECoG Platinum Electrode Grid*

Frederick W. Tidwell III, *Midshipman, USN*

***Abstract*— Implementing a brain-machine interfacing (BMI) Classifier that can distinguish between classes in future data sets with a low error rate can prove to be difficult. In this study, different classifiers and features were selected in order to determine if a patient with an 8x8 electrocortigraphic (ECoG) platinum electrode grid is imagining movement with their finger or tongue. To develop this classifier using MATLAB, the training data was filtered to extract the characteristics of the mu wave (7.5 to 13 Hz). Cumulative Line Length of each channel was selected as the training feature set. The feature set was normalized and reduced using a sequential forward feature selection through MATLAB. With the reduced feature set, the filtered training data and labels were ran through a Linear Support Vector Machine (SVM). This approach was analyzed with test data from the next week. The classifier achieved a 10% error.**

I. Introduction

Students in the EE486E (Brain-Machine Interfacing) course were provided ECoG data that was from a patient instructed to execute imagined finger or tongue movements after seeing an image of a finger or a tongue on a screen. Data collection started 0.5 seconds after the image appeared to cut any visually evoked action potentials propagated from the Occipital Lobe. The 8x8 cm ECoG grid covered the contralateral motor cortex, but also some surrounding cortical areas. There were 278 three second trials conducted and voltages from all channels were sampled at a rate 1 kHz.

The data set given contained a 1x278 vector, "Y", that held the label 1 or -1, which told if the subject was imagining finger or tongue movement, respectively. The other given information was a 278x64x3000 array that held the 3000 collected voltages (in microvolts) of each of the 64 channels from all trials.

Past research has shown that imagined movements, such as closing your hand, modulates electrical activity in the brain's mu band, according to Dr. Nitish V. Thakor, the director of the Laboratory for Neuroengineering at Johns Hopkins University. [1] In addition, the mu band has been linked with the "mirror neuron" system, which plays a major role in the ability to imitate behaviors and actions. [2]. Since influences in this band region has had great success with ECoG robotic control, the mu band will most likely be beneficial in making decisions between different areas of the body, such as an imagined finger or tongue movement.

II. Method

A. Filtered Mu Wave

The first step in modeling the BMI Classifier is to determine feature extraction. In this classifier, the data set went through a Band-pass filter with a passband range from 7.5 to 13 Hz. This frequency range is noted for carrying the Alpha and mu brain waves. According to the director of the Laboratory for Neuroengineering at Johns Hopkins University, Dr. Nitish V. Thakor,

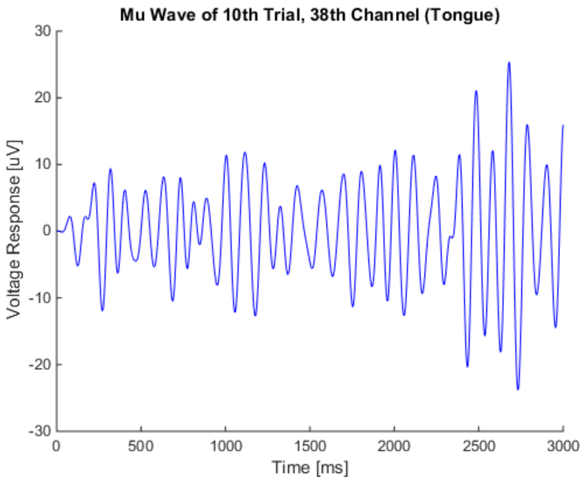
In MATLAB, the filter_IF function was implemented to create the Band-pass filter. This filter was given to students taking Modern Communication Systems in order to design a Superheterodyne receiver. Filter_IF was a simple way of using a Butterworth filter where the inputs are the sampling frequency and the lower and upper -3dB cutoff frequencies. Figure 1 and 2 illustrate the differences between the mu waves of an imagined finger or tongue movement, respectively.

Figure 1. Imagined Finger Mu Wave

*Research supported by EE486E course.

F. A. Author is with the United States Naval Academy, Annapolis, MD 21412 USA (corresponding author to provide phone: 352-302-2050; e-mail: fred.tidwell.03@gmail.com).

Figure 2. Imagined Tongue Mu Wave



B. *Feature Extraction – Line Length*

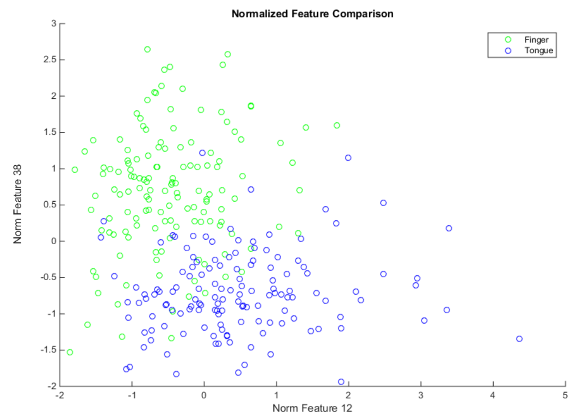
Line Length of each channel was selected after comparing the two mu waves in Figures 1 and 2. To implement this, a nested for loop was scripted that would execute the line length of all 64 channels and 278 trials, resulting in a 278x64 training feature array. The data was then normalized by subtracting by its mean and then dividing by the standard deviation of the filtered mu wave.

Because the electrode grid is covering the entire right motor cortex and other cortical areas, it is assumed that many of these channels will not have an effect on action potentials related to imagined movements of the finger or tongue and may act as noise to the BMI. The decided way to eliminate the noisy channels was to do forward feature selection with MATLAB's sequentialfs function. Sequentialfs does 10-fold cross validation with a user-inputted function as its criterion and stops when the removal of features has no improvement on the criterion. In our classifier, the criterion function was a created method that computed the number of errors using a Linear Support Vector Machine (SVM) classifier. 200 of the 278 trials were used as the training features and the remaining 78 as the test features. This function was executed several times to determine which normalized feature subset resulted in the lowest error through cross validation.

C. *SVM and K-Fold Cross Validation*

Linear SVM is a great tool to implement binary classification. When data sets seem to have a linear trend, a SVM can boost performance by creating a decision margin where the hyperplane decision boundary is fitted with a margin to each side. Figure 3 is an example of Channels 12 and 38's line length implemented with the green circles representing imagined finger movements and blue representing imagined tongue movements.

Figure 3. Line Length Comparison of Two Features



With more features, a hyperplane will have a smoother fit than that of just two features.

Since the student has no record of test data to determine error rate, the training data must be used to estimate error. The best approach is to use K-Fold Cross Validation with K equal to 10. This takes a tenth of the training set to be used as the test set, computes error, and then chooses another tenth to be the test set. This is iterated ten times. The feature subset that has on average the best error rate will be chosen for the final classifier.

D. *Final Classifier*

In the final BMI classifier, the test data will be inputted into the function and the predicted labels will be outputted. The classifier filters the data's mu band, extracts the selected line length features chosen from the sequentialfs function previously, normalizes and then runs normalized test features through the linear SVM structure that was saved after doing k-fold cross validation.

III. Results

On my first attempt at solving this problem, I executed the sequentialfs and did 10-fold validation 200 times to see what the average error rate would be. Figure 4 shows my results for different classifiers that I implemented. Figure 5 is a close up on Logistic Regression and Linear Discriminant Analysis performance.

Figure 4. Mean Error Rate for Sequentialfs Feature Selection for Different Classifiers

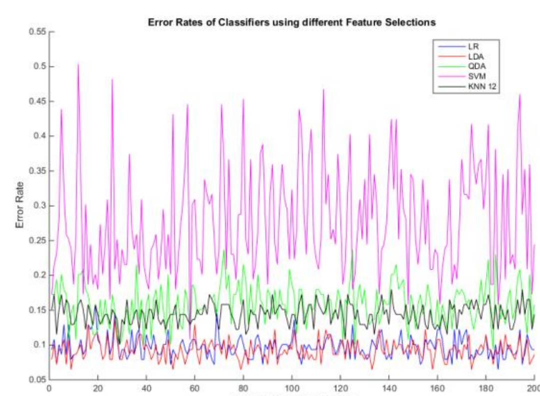
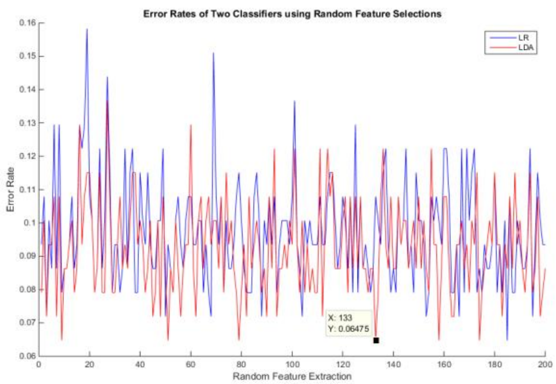


Figure 5. Figure 4 Close-Up on Logistic Regression and Linear Discriminant Analysis



Choosing Linear Discriminant Analysis (LDA) for the feature selection found on the 133th iteration of sequentialfs resulted in the lowest error rate. When building the LDA Classifier, I achieved a test error rate of 11%.

Later on I realized my support vector machine error was so high because it was executing in a Gaussian radial basis mode. I then changed it to a linear SVM and was able to drop my test Error Rate down to 10%.

Because some of the channels interfered with the desired data, as mention earlier, we cropped out 56 of the channels. The remaining 8 channels included channels 1, 3, 12, 18, 38, 40, 52, and 58.

Before you begin to format your paper, first write and save the content as a separate text file. Keep your text and graphic files separate until after the text has been formatted and styled. Do not use hard tabs, and limit use of hard returns to only one return at the end of a paragraph. Do not add any kind of pagination anywhere in the paper. Do not number text heads-the template will do that for you.

IV. Discussion

This classifier performed at an optimum error rate. The top two procedures that were needed to drop the error rate to this level was through filtering out the underlying mu wave and discarding unneeded channels.

As a whole, the classifier filtered out the mu wave for each channel, used the line length of each channel as the feature set, normalized, selected the optimum channels through sequential feature selection, and the remaining normalized features were inputted into a linear SVM classifier.

Future work on this project is to achieve a 9% error rate or less. I believe this can be done by also extracting data from a higher frequency band, the gamma band, from the range of 70 to 150 Hz. This is similar to what Johns Hopkins did in able to control robotic grasping according to Dr. Thakor. [1]

Appendix

The Appendix contains attached MATLAB scripts at the end of the paper.

Acknowledgment

I would like to thank Assistant Professor Justin Blanco for giving this course the opportunity to work with data sets that were actually collected from real BMI devices and not simulated data. This course will most definitely be useful as I pursue a graduate education in the near future.

References

[1] N.V. Thakor, (2014, August). How to Catch Brain Waves in a Net [Online]. Available http://spectrum.ieee.org/biomedical/bionics/how-to-catch-brain-waves-in-a-net.html

[2] L. Gabriel, (2011, June). Brain Wave Basics – What You Need to Know about States of Consciousness [Online]. Available http://thoughtmedicine.com/2011/06/brain-wave-basics-what-you-need-to-know-about-states-of-consciousness/