Finger Flexion Challenge Using ECoG Data

Zhiyu Kuang, Linran Wei, Yuanchao Zhang

November 8, 2024

Abstract

The age of machine learning opens up new opportunities to explore recorded brain signals in unprecedented ways. For example, newly emerged computational tools have paved novel paths in understanding Electrocorticography (ECoG) signals, which had been challenging to interpret due to their high spatial and temporal resolutions. This report focuses on several methods that used ECoG to predict finger flexion GLOVE. After choosing appropriate pre-processing, regression algorithm, and post-processing, **a correlation of 0.5610** between the predicted flexion and true GLOVE data was achieved using gradient boosting. 1 will give an overview of the prediction pipelines. 2 provides detailed descriptions of each step. 3 presents the results of the proposed methods. 4 discusses other approaches the team experimented with and the limitations of the work.

1 Overview

Finger flexion is a fundamental yet intricate motor action critical for our daily lives that can indicate seizures of patients with epilepsy or Parkinson disease. However, the cortical areas controlling the fingers can overlap with other movements, making it challenging to decode the signals that correspond to the flexion of certain fingers. Additionally, ECoG data, collected invasively with multiple electrodes, is not only prone to be noisy but also has high spatial and temporal resolutions. The high spatial resolution makes it difficult to isolate specific signals related to each finger's movement while high temporal resolution requires precise time-series analysis. Overall, using ECoG data to decode the finger movement remains a nontrivial challenge.

To pre-process the ECoG signal, we excluded channels with outlier data and performed filtering and normalization. Then, the pre-processed signals are rearranged using sliding window method. Within each window, 6 types of features were extracted for each channel, generating a feature matrix. After experimenting with different approaches, we found **GradientBoosting Regressor** yielded the highest correlation. The predictions from GradientBoosting Regressor were filtered through a 1-D Gaussian filter and underwent a rescaling procedure. Finally, the post-processed results were compared with true GLOVE data and evaluated using Pearson correlation. The best correlation from our pipeline is **0.5517** based on the leaderboard results. The overall workflow is visually represented in Fig.1.

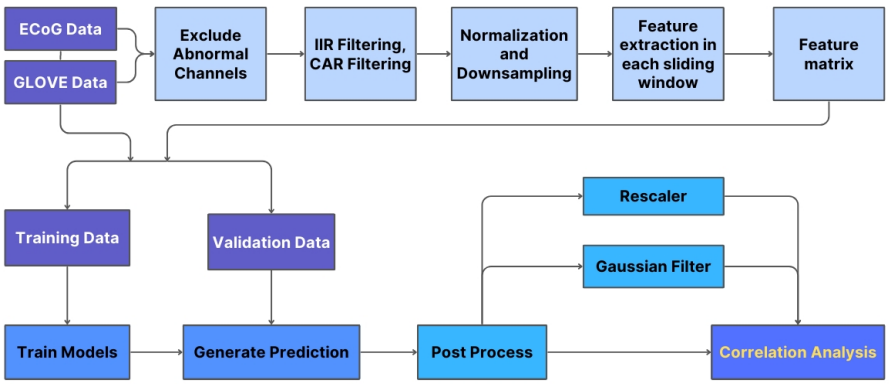


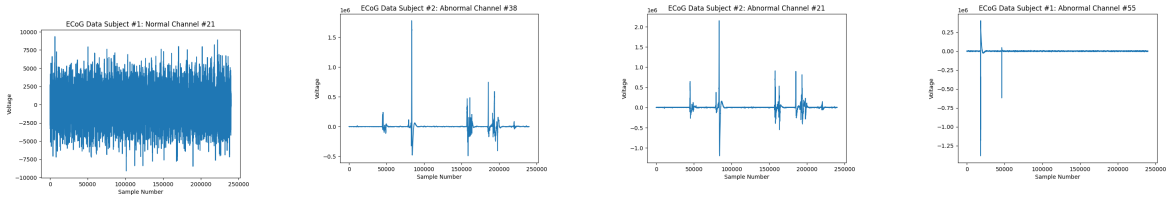
Figure 1: Workflow diagram of the algorithm

2 Algorithm

2.1 Pre-Process

The dataset was collected from three subjects with ECoG of 62, 48, and 64 channels[4, 5]. Based on the data description given in the class recitation, channel 55 from subject 1 and channel 21, 38 from subject 2 are abnormal channels. It can be seen from Fig.2, there is a drastic activity difference between the normal and abnormal channels, thus the necessity of removing outliers is justified in this case.

Though the given signal is filtered (0.15-200 Hz) [4, 5], to ensure the quality, the team chose to filter the data again to reduce noise. The filter was chosen to be a second-order Infinite Impulse Response (IIR) notch filter which is suitable for eliminating specific frequencies from a signal. IIR uses both past output and past input values to compute its current output. Based on [1], we tried to eliminate 60 Hz noise induced from measurement.



(a) Normal channel

(b) Abnormal channel 38 subject 2

(c) Abnormalchannel 21 subject 2

(d) Abnormalchannel 55 subject 1

Figure 2: Comparison between normal and abnormal channel neural activities

Following the initial bandwidth filtering, the team performed an additional Common Average Reference filtering (CAR). An essential assumption of applying CAR is that ECoG signals are contaminated by several artefacts that are common across all the electrodes. CAR calculates the mean signal across all electrodes at every timestep and creates a time series that represents the common signal component across all channels. Then, the algorithm subtracts this average signal from the original signal, expressed in Eq.(1).

$$y_i[t] = x_i[t] - \frac{1}{N}\sum_{n=1}^{N} x_n[t] \tag{1}$$

where $y_i[t]$ is the CAR filtered signal at channel i and time t, $x_i[t]$ is the original signal at channel i and time t, N is the total number of channel, and $\frac{1}{N}\sum_{n=1}^{N} x_n[t]$ represents the average signal across all channel at time t. This technique potentially increases signal-to-noise ratio.

Before extraction the features, the signal is further normalized using StandardScaler to make its values more consistent for each channel. GLOVE data is downsampled to 25 Hz since we want to recover the original signal collected from the digital gloves, according to [4].

2.2 Feature Selection

Choose the appropriate features are essential to the robustness of the model. After experimenting with different features, the team included the following to train the model: 1. line length feature; 2. relative band power of a specified frequency band; 3. average square voltage of a particular frequency range; 4. total spectral power; 5. area; 6. average variance and mean over time.

Line length feature quantifies the complexity or the amount of change in the signal over a given period. Area evaluates the magnitude of the signal over time. Variance and mean voltage are also evaluated in the time domain, giving information about how the signal distributes.

Relative band power for a specific frequency gives insights into understanding how the specific frequency range contributes to the total power of the signal. Power Spectral Density (PSD) is first calculated using Welch's method for each channel. Then, PSD is integrated over a specific frequency band, and the ratio of that band over the complete bandwidth is the feature. As suggested in checkpoint 1, the team chose 5-15 Hz, 20-25 Hz, 75-115 Hz, 125-160 Hz, 160-175 Hz for relative band power.

The team also included total spectral power as a feature. In general, it demonstrates the sum of energy across all frequencies which might potentially provide a broad measure of the neural dynamics when moving different fingers.

The average squared voltage of a particular frequency range is another simple but useful feature. It represents amplitude characteristics and dynamics of a specified frequency range. Based on [2] and some trial and error, the team found the most appropriate frequency range for this ECoG data to be: 1-30 Hz, 30-50 Hz, 50-80 Hz, 80-120 Hz, 120-200 Hz. Notice that this frequency ranges differ from that of relative band power.

Therefore, there are 15 features (6 types) in total, containing information on both time-domain and frequency-domain.

2.3 Feature Matrix

To generate features that carry the dynamic changes of the original data, the team chose to use the sliding window method where signals are segmented using different sliding windows aligned at the left side. The window length is 80ms with a 40ms window overlap to keep the consistency with the downsampled GLOVE data. The features are calculated for each channel within each window – every row of the matrix. The resulting feature matrix for one window of a subject has the shape of $(N \times (15 \times c))$ where N is the number of windows and c is the number of channels. As shown below, $x^n_{i,j}$ is the value of the j^{th} feature of i^{th} channel in the n^{th} window:

$$\mathbf{F} = \begin{bmatrix} x^1_{1,1} & x^1_{1,2} & \dots & x^1_{1,15} & x^1_{2,1} & \dots & x^1_{c,15} \\ x^2_{1,1} & x^2_{1,2} & \dots & x^2_{1,15} & x^2_{2,1} & \dots & x^2_{c,15} \\ \vdots & \vdots & \vdots & \vdots & & & \\ x^N_{1,1} & x^N_{1,2} & \dots & x^N_{1,15} & x^N_{2,1} & \dots & x^N_{c,15} \end{bmatrix}$$

2.4 Regression Algorithm

After creating the feature matrix, it is input into the training model – gradient boosting. This algorithm is an ensemble technique that captures nonlinearity, capable of handling large features and is robust to overfitting. Gradient boosting improves its model performance based on previous ones. Starting with a weak constant model, at each step, the model calculates the negative gradient,r_{im}, of the loss function with respect to the predictions. Then, the residual r_{im} is used to train the tree $h_m(x)$ through splitting where each tree selects the features that minimize Friedman-MSE for the two splitting nodes.

After fitting the model, the optimal step size is calculated to be γ_m and the final model is generated according to:

$$F_M(x) = F_0(x) + \nu \sum_{m=1}^{M} \gamma_m h_m(x) \tag{2}$$

where ν is the learning rate. When inputting test sets into $F_M(x)$, the prediction is produced.

2.5 Post-Process

Before calculating the Pearson correlation between the prediction and true GLOVE data, the results are further post-processed using 1-D Gaussian filter to reduce noise and fluctuations in the prediction. It is reasonable to assume that a time series data contains some sorts of continuity in time, thus applying the smoothing operation can help the model less sensitive to small variations and emphasize the more significant underlying patterns.

This process results in a new data series where each point is a weighted average of its neighbors, and the weights are determined by the Gaussian function. Such a method effectively smooths the data. The effect of the Gaussian smoothing can be seen from 3. In the bottom graph, the blue line represents the original and unfiltered prediction which appears to be noisy and ambiguous while the green line after the Gaussian filtration is more similar to the GLOVE true data. The result demonstrates that the Gaussian filtering improves the quality of the prediction. The sigma of each Gaussian filter is ideally different for each prediction, but in this experiment, the sigma is set to $\sigma = 4$ for all filters based on several tryouts.

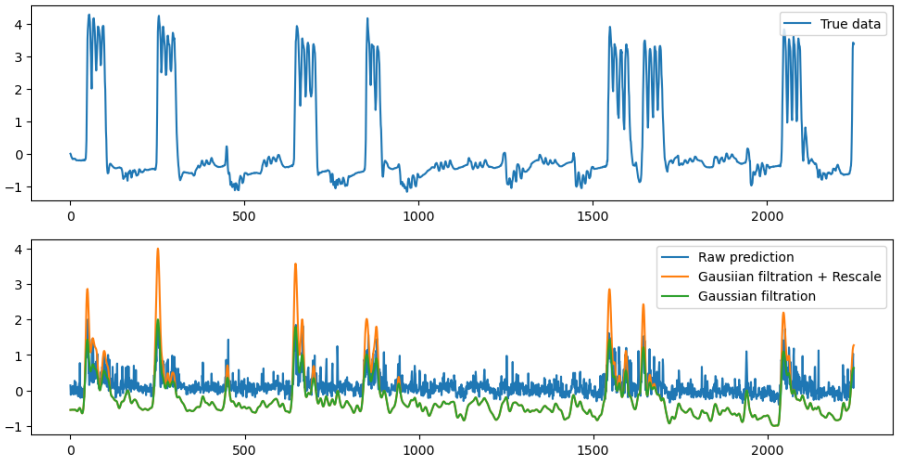


Figure 3: Comparison among true GLOVE data(upper graph); unfiltered(lower blue); Gaussian-filtered (lower green) and Gaussian and rescaled (lower orange) prediction.

A second post-process step was taken to scale the filtered prediction of three subjects on individual cases since we found that subject 2 tends to have more outlier data: (1) The prediction of Subject 1 was scaled using the StandardScaler; (2) The Prediction of Subject 2 was scaled with RobustScaler within the quantile range of (25, 75); (3) The prediction of Subject 3 was scaled using MinMaxScaler within the range of (-1, 2). For outputs that are greater than 0, they are further re-scaled to twice its values, making the positive prediction data (ie. finger movement) more distinguishable from the resting data. The range of the scaler and the rescaling factors are hyperparameters that are chosen after different experiments. The rescaled prediction is demonstrated in the lower graph of Fig.(3) in the orange line. Compared to only Gaussian-filtered data, the finger-moving phase stands out more from the resting stage.

3 Results

To have an idea of our model using raw_training_data file, we split the dataset into training and validation data with a ratio of 8:2. After the pre-processing, training, and post-processing stages, the predicted data is compared with true GLOVE data using Pearson correlation.

Gradient Boosting Regressor with n_estimators=100 generates a correlation of **0.5610** with leaderboard data. Our algorithm has a high true positive rate of detecting movement for most of fingers, shown in Fig.(4). This can be beneficial to the BCI application if it is designed to help with detecting the seizures of patients with finger movements.

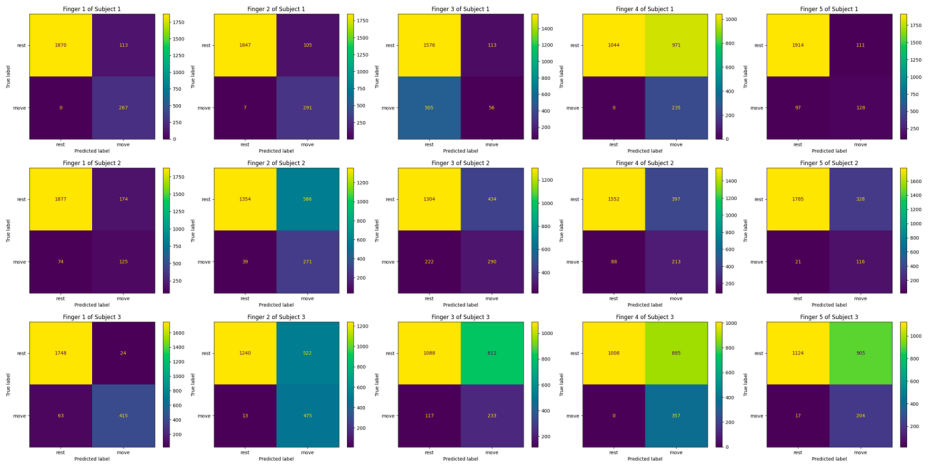


Figure 4: Confusion matrix of movement detection for each finger of each subject. The movement of finger is defined as a period with a large amount of higher values than threshold.

4 Discussion

4.1 Other Approaches

Other approaches include linear regression, random forest, CNN-LSTM [6], and Multilayer Perceptron (MLP). The results of three machine learning models are listed in Table 1 .

Linear regression produced a overall correlation is 0.47086. However, since linear regression is a relatively simple network with linear dynamics, it might not be capable of capturing the non-linearity of the data.

Model	Subject 1	Subject 2	Subject 3	Overall mean
Linear regression	0.349	0.396	0.668	0.471
Random Forest	0.484	0.494	0.662	0.547
Gradient Boosting	0.458	0.510	0.699	**0.556**

Table 1: Results of three machine learning algorithms using validation set split from raw training data

Random forest is similar to gradient boosting but different in a few ways. Most importantly, gradient boosting grows trees sequentially while random forest starts by growing trees independently based on subsets of features. The result of the random forest is around 0.54. One advantage of random forest is its computational efficiency compared to the gradient boosting, but our team chose gradient boosting based on performance [3].

Our team also experimented with CNN-LSTM and MLP because deep learning methods tend to do well with highly nonlinear dynamics. We tried to train the model on features and raw data separately. Both cases yielded poor results. We hypothesized that it might be due to the inappropriate structure of the models, and the inadequate amount of data samples.

4.2 Further Improvements

One limitation of our method is the manual selection of several hyperparameters. In the post-process of deciding the σ for the Gaussian filter and rescale factor of movement – in this experiment, those hyperparameters are largely dependent on manual tryouts. Post processing is crucial to our final algorithm, so tuning them more systematically and automatically might be more beneficial.

To discuss the reasons for excluding the activities during the fourth finger movement, our team hypothesizes that the ability to move the fourth(ring) independently is the hardest among the five fingers. Particularly, when we try to move the fourth(ring) finger, we usually need help from the third finger(middle) and fifth finger(litter), so it is not surprising that the fourth finger is highly correlated with the third and the fifth finger.

Overall, our team enjoyed exploring different methods and especially the excitement when our findings made the correlation higher and we were proud of ourselves for being able to aggregate all our knowledge together to overcome this challenge. We also recommend that in the future, the class should provide a list of more reference papers to save them some time. Overall, this has been an invaluable experience for our team.

References

[1] James M. Fiore. *11.8: Notch Filter (Band-Reject) Realizations*. Engineering LibreTexts. June 24, 2020. URL: https://eng.libretexts.org/Bookshelves/Electrical_Engineering/Electronics/Operational_Amplifiers_and_Linear_Integrated_Circuits_-_Theory_and_Application_(Fiore)/11%3A_Active_Filters/11.08%3A_Notch_Filter_(Band-Reject)_Realizations (visited on 04/23/2024).

[2] J Kubánek et al. "Decoding flexion of individual fingers using electrocorticographic signals in humans". In: *Journal of Neural Engineering* 6.6 (Dec. 1, 2009), p. 066001. ISSN: 1741-2560, 1741-2552. DOI: 10.1088/1741-2560/6/6/066001. URL: https://iopscience.iop.org/article/10.1088/1741-2560/6/6/066001 (visited on 04/23/2024).

[3] Tomonori Masui. *All You Need to Know about Gradient Boosting Algorithm Part 1. Regression*. Medium. Feb. 18, 2024. URL: https://towardsdatascience.com/all-you-need-to-know-about-gradient-boosting-algorithm-part-1-regression-2520a34a502 (visited on 04/23/2024).

[4] Kai J. Miller and Gerwin Schalk. *Prediction of Finger Flexion 4th Brain-Computer Interface Data Competition*. URL: https://www.bbci.de/competition/iv/desc_4.pdf.

[5] Gerwin Schalk et al. "Decoding two-dimensional movement trajectories using electrocorticographic signals in humans". In: *Journal of neural engineering* 4 (Oct. 2007), pp. 264–75. DOI: 10.1088/1741-2560/4/3/012.

[6] Ziqian Xie, Odelia Schwartz, and Abhishek Prasad. "Decoding of finger trajectory from ECoG using deep learning". In: *Journal of Neural Engineering* 15.3 (June 1, 2018), p. 036009. ISSN: 1741-2560, 1741-2552. DOI: 10.1088/1741-2552/aa9dbe. URL: https://iopscience.iop.org/article/10.1088/1741-2552/aa9dbe (visited on 04/23/2024).