BE521: Final Project

Spring 2019

12 April 2019

1 Project overview

This final project involves predicting finger flexion using intracranial EEG (ECoG) in three human subjects. The data and problem framing come from the 4th BCI Competition. For the details of the problem, experimental protocol, data, and evaluation, please see the original 4th BCI Competition documentation (included as separate document). The remainder of the current document discusses other aspects of the project relevant to BE521.

2 Important deadlines

Date	Requirement	Points
Wednesday , April 17, 11:59pm	team registration	5
Wednesday , April 24, 11:59pm	pass testing set checkpoint 1	20
Wednesday , May 1, 11:59pm	pass testing set checkpoint 2	15
Friday , May 3, by 11:59pm	end of competition, submit algorithm (Canvas)	15
Monday, May 6, 12:00pm	hand in final report	60
Monday, May 6, 12:00-2pm	competition results in Moore 216	Pizza/Prizes

The grading is structured so that you do not have to "hit this one out of the park" to do well, but going the extra mile is definitely rewarded. We want you to show what you've learned this semester, and to have some fun!

3 Dataset

Please refer to the accompanying *Competition Description* document to learn about the datasets (however, ignore the section regarding MAT-files) The dataset has been uploaded to the IEEG Portal and can be accessed as follows:

- Subject 1
  - `I521_Sub1_Training_ecog` - Training ECoG
  - `I521_Sub1_Training_dg` - Training Data Glove

- `I521_Sub1_Leaderboard_ecog` - Testing ECoG

- Subject 2
  - `I521_Sub2_Training_ecog` - Training ECoG
  - `I521_Sub2_Training_dg` - Training Data Glove
  - `I521_Sub2_Leaderboard_ecog` - Testing ECoG
- Subject 3
  - `I521_Sub3_Training_ecog` - Training ECoG
  - `I521_Sub3_Training_dg` - Training Data Glove
  - `I521_Sub3_Leaderboard_ecog` - Testing ECoG

Your task is to develop an algorithm to use the ECoG to predict finger flexion that is captured by the Data Glove.

4 Competition

Registration, submission and the leaderboard will be at `http://be521.pythonanywhere.com/`

4.1 Registration

1. You must form a team of two or three people. If you do not have a team, use Piazza, come to office hours, or let us know and we can randomly assign you to a team.
2. Each team will register by filling out a registration form at the website above. Please use your Penn email addresses. Spaces and other normal characters are allowed in the team name, but be creative!

4.2 Leaderboard

1. You will use the "Testing ECoG" data for each subject (`I521_Sub*_Leaderboard_ecog`) to create your Data Glove predictions to submit to the leaderboard. Submissions can be made on the website. The leaderboard will show whether you've passed the checkpoints as well as the current ranking.
2. Any member of the team can make submissions to the leaderboard using the registered email.
3. Leaderboard submissions must be: Matlab *.mat file with one variable: `predicted_dg`. This variable is a 3 x 1 cell array, where `predicted_dg{i}` corresponds to the predictions for subject i. [1]. Thus, `predicted_dg{1}` is a 147,500 x 5 dimensional matrix for the 147,500 time points and 5 fingers predicted from subject 1's testing ECoG.
4. The submission results will be refreshed every 6 hours at the beginning of the competition, and more frequently as deadlines approach. This is to help reduce overfitting to the supplied test set.

[1] These submission variables are slightly different from those required in the original BCI Competition. Be sure to follow our variable specifications.

4.3 Evaluation

Performance will be measured by the average correlation coefficient across the 3 patients on fingers 1, 2, 3, and 5. [2]

4.3.1 Checkpoints

We will use your maximum achieved correlation to evaluate checkpoints.

1. Checkpoint 1 corresponds to achieving an average testing set correlation coefficient of $\mathbf{r} \geq \mathbf{0.33}$. The points for this checkpoint are all-or-nothing, i.e., you get all points if you pass it and 0 points if you do not.

2. Checkpoint 2 corresponds to achieving an average testing set correlation coefficient of $\mathbf{r} \geq \mathbf{0.45}$. The points for this checkpoint are broken down as follows: 5 points for $\mathbf{r} \geq \mathbf{0.40}$ and the remaining 10 points for $\mathbf{r} \geq \mathbf{0.45}$.

4.3.2 Final ranking

We will run your model on an independent test set for the final rankings and results will be announced at the final presentation.

4.4 Prizes

The prizes for the top three winners are

1st place: an automatic A in the **class** for each member of the team

2nd place: +5 points on the final **class grade**

3rd place: +3 points on the final **class grade**

Other awards will be given for creativity and the "Medal of Valor Award" for that team which kept trying in the face of adversity (however we define it).

5 Deliverables

In addition to the registration and checkpoints above, you will have to submit your algorithm and report. Only 1 of each needs to be submitted per team.

1. **Final Submission Script and Auxiliary files**
   (due ON CANVAS 1 hour after competition closes)
   Your team will have to complete and submit the provided `make_predictions.m` script that takes in all subject's ECoG and outputs a predicted data glove for each subject. See `make_predictions.m` for more details. In this script, you should load your models from a `.mat` containing your model(s) for each subject, do the necessary testing, and return the predictions. You should not do any model training. Any libraries (e.g. LibSVM) that you use should also be included.

[2] We do not test on the 4th finger because it is highly correlated with the 3rd finger

Any late submissions or errors that are encountered when the TAs run each team's code will result in automatic disqualification of the team from the competition. Yes, this means that even if you come in 1st place, if you submitted late or we had problems running your code, you will not get an automatic A. TA's will provide a test script and hold a session where you can make sure your algorithm runs.

2. **Final Report**
   Due: Monday, May 6th, 12:00 PM at the final meeting.
   NO LATE SUBMISSIONS ACCEPTED
   Teams must hand in a final report in person, 1 report per team. Make sure your team members are listed. The report should be 3-5 pages, 1.5 spaced (or similar) and should describe your final algorithm in detail. In particular, your report must have the following

   - a summary paragraph describing your final algorithm (5 pts)
   - a step-by-step, detailed explanation of your final algorithm (15 pts)
   - a flow chart summarizing the general steps of your algorithm (5 pts)
   - at least one interesting figure that illustrates part of your algorithm or motivates a particular decision you made along the way (5 pts)
   - a discussion of what methods you tried that did not work (10 pts)
   - some thoughts about why the fourth (ring) finger's flexion was generally so correlated with the third (middle) and fifth's (little) (5 pts)
   - a conclusion about your overall experience with the project and how well you felt you did on it (5 pts)
   - a references section listing the papers and other resources you used (5 pts)
   - an appendix with the code used in your final version (i.e. please do not include all the code you ever wrote, just what your finished solution used) (5 pts)

   In addition to these points, the report will be graded on the clarity of the writing and thoroughness of the explanation.

6 Honor code

While you are allowed to consult other sources, your team must write your own code. External libraries such as LIBSVM and all Matlab toolboxes are allowed.

7 Guidance

In this section, we first try to give you some general advice and then a description of a straightforward method that should get you well on your way to passing at least the first checkpoint.

General advice

1. Get started early! The sooner you pass the checkpoint(s), the sooner you can relax a little bit and have fun with the competition. Submit early and often as you make progress. Do not wait until the last minute to start submitting results.

2. Figure out a good system within your team to share and (if possible) version-control your code. Emailing versions back and forth can quickly become cumbersome and confusing. If you really want to do it right, you should set up a shared repository using (hyperlinks underneath)

   - SVN via CETs at UPenn
   - Github
   - BitBucket

   a repository not only provides a central home for your code, but it also provides easy version control, so you can look back in time and/or roll back the state of your code if need be.

   A simpler yet still good alternative would be to use some sort of shared network drive or folder so that at least your code is in one place. Free software like Dropbox is easy and even provides a bit of version control (through their website) if you need it. But, this can also lead issues such as conflict files when two people are working on the same file at the same time.

3. Document your code well (i.e. comments!). Good documentation will be invaluable to the rest of your team. It is also imperative when you're always going back to old code and trying to improve it. This habit will always serve you well. If your code is unreadable we will deduct points.

4. Organize your code well. Do not simply put it all in one big script. Split out small parts (like feature extraction, training, and making the predictions) into functions that you can easily call for each subject and/or each finger. You may want to have one main script (with code sections) which you can run that goes through all the steps, from loading the data to making the predictions. Talk to TA's if you'd like advice on code organization.

5. Save the results of intermediate processing if it makes sense. For example, you may want to save the results of your feature extraction (which can be time-intensive, depending on your features) to a file so that you don't have to do it again for the near future. That way you can easily share with your other team members and/or come back to your work another day (or Matlab session) and pick up in the middle where you left off.

6. Design. Build. Test. Iterate. Do not try to solve this problem all at once. Start simple and get something working, even if it's not working as well as you want. Once you have your general algorithm flow down (scripted!, see previous point) it's much easier to improve small pieces of it. You may want to keep a notebook or text-file log of current attempts and final performance metrics so it's easier to compare "versions" of your algorithm. Trust us, all of your ideas will not necessarily make your algorithm better, so you will want some quantitative way to compare versions.

7. Cross-validation is your friend. If you're doing any learning that has some tunable parameters involved, you will probably want to cross-validate. Even if you're not tuning parameters, cross-validation will give you a better sense of what your generalization performance (i.e., testing performance) will be. Training performance can be deceptively good.

8. Be creative and clever. Remember, everyone else is probably thinking of the same obvious things to do as you. Keep in mind some of the areas/skills you have at your disposal, including (but by no means limited to)

   - neuroscience: timescales of motor planning
   - feature extraction: sliding windows, interpolation
   - signal processing: time and frequency-domain features, convolution, smoothing
   - unsupervised learning: clustering, dimensionality reduction (e.g., PCA)
   - supervised learning: classification
   - prediction: linear regression/prediction

   The best methods will almost certainly combine a number of these tools.

9. Consult outside references. You're probably not the first person who has tried to do motor prediction from ECoG data (although there aren't a ton!), so see what other people have tried as a good grounding and/or jumping off point. Some recent papers of Gerwin Schalk (who gave the P300 lecture and was one of the designers of the BCI Competition) might be a good place to start.

10. Start working on a small chunk of data, perhaps a half or even a quarter of the training data for one subject. Things will go faster that way when you're getting your algorithm up and running. Once you think you have a reasonable algorithm, then expand what data you are working with.

11. You may find it useful to sometimes have multiple Matlab sessions running. Running Matlab in the terminal is much more lightweight than a big graphical session and so might be an option if your OS is having trouble with multiple graphical sessions.

12. You may also find it useful to print status messages to the console (using `disp` or `fprintf`) to tell you what your code is currently doing, e.g. where it is in a loop. For example, when our code is doing feature extraction, it prints out the current subject ("Subject 1", "Subject 2", etc) and then a little dot (.) after processing each channel. Such feedback can sometimes make longer computations more bearable because you have a better idea of where you are in them and when they will finish.

A Simple Method To help get you on your way with this problem, below we outline a simple method adapted from Kubanek *et al.* (J Neural Engineering, vol 6, 2009).

For each subject individually,

1. We used a moving window 100 ms in length with 50 ms overlap and extracted the same 6 features over each of the EEG channels. The features were the average time-domain voltage,

and the average frequency-domain magnitude in five frequency bands: 5-15 Hz, 20-25 Hz, 75-115 Hz, 125-160 Hz, 160-175 Hz. (N.B., the `conv` and `spectrogram` functions, respectively, were very helpful for this.) Thus, the total number of features in a given time window was $62(5+1) = 372$ for a subject with 62 EEG channels.

2. We downsampled the dataglove traces (using the `decimate` function) so that each sample was separated by 50 ms, to keep them on the same time scale as the features.

3. We used linear prediction (linear regression) to predict each (downsampled) finger flexion from all the EEG features from the previous three time windows (so 150 ms lag). This process was basically the same as what you did in the motor prediction homework. Note, for matrix $\boldsymbol{X}$ representing the EEG data (same as the $\boldsymbol{R}$ matrix from the motor-prediction homework) and target matrix $\boldsymbol{Y}$ (where each column is a different finger flexion trace), the formulae for the filter coefficients is still the same: $\boldsymbol{\beta} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{Y}$. The weights $\boldsymbol{\beta}$ are now a matrix instead of a vector, but that means that since we have the same $\boldsymbol{X}$ for each finger, we can predict all finger positions at once with $\hat{\boldsymbol{Y}} = \boldsymbol{X}\boldsymbol{\beta}$.

4. We then interpolated the prediction using a cubic spline (see the `spline` function) back up to the original 1000 Hz sampling frequency, making sure that the first and last points in the data interpolation were values we know (i.e., not values we needed to interpolate, since the cubic spline often blows up at the edges if not constrained). The interpolation was zero-padded and the beginning and end to time-align with the original flexion trace.

This method achieves an average correlation coefficient of at least $r = 0.33$ on the testing set.

Prediction of Finger Flexion
4^{th} Brain-Computer Interface Data Competition

Kai J. Miller[1,2] and Gerwin Schalk[3,4,5]
[1]*Department of Physics, University of Washington, Seattle, Washington*
[2]*Program in Neurobiology and Behavior, University of Washington, Seattle, Washington*
[3]*Wadsworth Center, New York State Department of Health, Albany, New York*
[4]*Department of Neurology, Albany Medical College, Albany, New York*
[5]*Department of Neurosurgery, Washington University in St. Louis, St. Louis, Missouri*
(Dated: June 11, 2008)

The goal of this element of the competition is to predict the flexion of individual fingers from signals recorded from the surface of the brain (electrocorticography (ECoG)). This data set contains brain signals from three subjects, as well as the time courses of the flexion of each of five fingers. The task in this competition is to use the provided flexion information in order to predict finger flexion for a provided test set. The performance of the classifier will be evaluated by calculating the average correlation coefficient r between actual and predicted finger flexion.

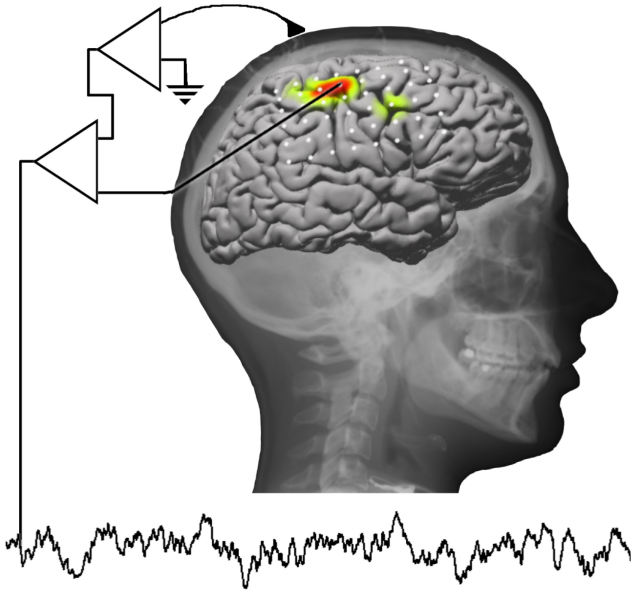
FIG. 1: ECoG signals in train_data(time,channel) and test_data(time,channel) were acquired from each electrode with respect to a scalp reference and ground before re-referencing with respect to the common average.

I. SUBJECTS

The three subjects in the data set were epileptic patients at Harborview Hospital in Seattle, Washington. Each patient had electrode grids placed subdurally on the surface of the brain for the purpose of extended clinical monitoring and localization of seizure foci. Each subject gave informed consent to participate in this study, which was approved by the internal review board (IRB) of Harborview Hospital. All patient data have been anonymized according to IRB protocol in accordance with HIPAA regulations.

II. RECORDINGS

Signals from the electrode grid were amplified and digitized using Synamps2 amplifiers (Neuroscan, El Paso, TX). The general-purpose BCI system BCI2000 [1] provided visual stimuli to the patient, acquired brain signals from the Synamps2 system, and also recorded the flexion of individual fingers (on the hand contralateral to the implanted grid) using a data glove (Fifth Dimension Technologies, Irvine, CA). BCI2000 stored the brain signals, the timing of stimulus presentation, and the flexion of each of the fingers in a data file. Data files were converted to Matlab format for this competition. Each patient had subdural electrode arrays (Ad-Tech, Racine, WI) implanted. Each array contained 48-64 platinum electrodes that were configured in 8x6 or 8x8 arrangements. The electrodes had a diameter of 4 mm (2.3mm exposed), 1 cm inter-electrode distance, and were embedded in silastic. Electrocorticographic (ECoG) signals (i.e., 62, 48, and 64 channels from subjects 1, 2, and 3, respectively) were acquired with respect to a scalp reference and ground (Fig. 1), band pass filtered between 0.15 to 200 Hz, and sampled at 1000 Hz.

III. EXPERIMENTAL PROTOCOL

The subjects were cued to move a particular finger by displaying the corresponding word (e.g., "thumb") on a computer monitor placed at the bed-side (Fig. 2). Each cue lasted two seconds and was followed by a two-second rest period during which the screen was blank. During each cue, the subjects typically moved the requested finger 3-5 times. This number varied across subjects and fingers. There were 30 movement stimulus cues for each finger (i.e., a total of 150 cue presentations and about 90-150 flexions of each finger); stimulus cues were interleaved randomly. This experiment lasted 10 minutes for each subject.

Subsequent offline analysis showed that ring (4^{th}) fin-

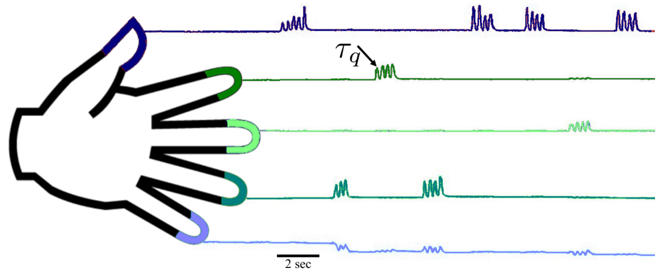

FIG. 2: Capturing individual finger flexion: each subject moved a finger several times in response to a visual cue. Finger flexion was recorded using a dataglove. The task in this competition is to use the provided ECoG signals to predict the flexion of each finger during the last 1/3 of the experiment.

ger movements were correlated with either middle (3rd) or little (5th) finger movements. Thus, while this ring (4th) finger position is included with the training data, it will not be used for evaluation purposes.

IV. DATA STRUCTURE

The data for each subject are contained in a separate MATLAB file that is named *"subX_comp.mat"* where *"X"* denotes the subject number. Each file contains three variables:

- *"train_data"* - this variable, in time × channels, gives the first 2/3 (6 min, 40s) of recorded ECoG signals (400,000 samples at 1kHz sampling rate per channel) from the specified experiment, for every channel.
- *"train_dg"* - this variable, in *time* × *finger* is the first 2/3 (6 min, 40s) of recorded finger position (thumb - index - middle - ring - little; 400,000 samples (super-sampled to 1 kHz) per finger) for the associated experiment.
- *"test_data"* - this variable, in time × channels, gives the last 1/3 (3 min, 20s) of recorded ECoG signals (200,000 samples at 1kHz sampling rate per channel) from the specified experiment, for every channel. These data will be used to predict the final 1/3 (3 min, 20s) of recorded finger position (thumb - index - middle - ring - little) for the associated experiment.

Please note that the channel order has been scrambled.

V. EVALUATION CRITERIA

Each participating group will submit three files titled *"sub1_eval"*, *"sub2_eval"*, and *"sub3_eval"*, corresponding to subjects 1-3, respectively. Each of these will contain a single variable, *"eval_dg,"* with dimensions 200,000 × 5:

- *"eval_dg"* - this variable, in time × channels, shall give the last 1/3 (3 min, 20s) of predicted finger flexion for each of the five fingers (thumb - index - middle - ring - little) for the associated experiment (200,000 samples per finger).

The evaluation criteria will be as follows: for each subject, the received variable *"eval_dg"* will be compared with the actual finger positions in *"test_dg,"* which we have retained. We will calculate the correlation coefficient r between the actual and the predicted finger flexions for each subject and finger. We will not calculate the correlation coefficient for the 4th (ring) finger, because the flexion of this finger was typically correlated with the flexion of the 3rd (middle) or 5th (little) finger. The variable for predicted position should still be of dimension 200,000 × 5 (i.e., the 4th column will not be used for evaluation). The final score will be calculated as the arithmetic mean of the 12 correlation coefficients (4 per subject, 3 subjects). The submission with the highest score wins the competition.

VI. SOME THINGS TO KEEP IN MIND...

- The lag between the dataglove position measurement recording and the amplifier measurement is 37ms (± 3ms, SEM). This is of the same order of granularity in the dataglove position (which was sampled at 25Hz - every 40ms).
- There is a characteristic delay between brain activity and resulting finger movement. You may want to take this into account.
- Please cite [2] when you use this dataset in a publication.

[1] Schalk, G., McFarland, D.J., Hinterberger, T., Birbaumer, N., and Wolpaw, J.R. BCI2000: a general-purpose brain-computer interface (BCI) system, *IEEE Trans Biomed Eng*, 51(6): 1034-1043, 2004.

[2] Schalk, G., Kubanek, J., Miller, K.J., Anderson, N.R., Leuthardt, E.C., Ojemann, J.G., Limbrick, D., Moran, D.W., Gerhardt, L.A., and Wolpaw, J.R. Decoding Two-Dimensional Movement Trajectories Using Electrocorticographic Signals in Humans, *J Neural Eng*, 4: 264-275, 2007.