BCI practical course : Imagined Movements

Jason Farquhar (coordinator)

<J.Farquhar@donders.ru.nl>

Loukianos Spyrou (Teaching Assistant)

<l.spyrou@donders.ru.nl>

Learning Goals – Imagined Movement

- Know how to:
  - Build a multi-block experiement
  - Collect labelled data during the callibration phase of an experiment for later classifier training
  - Train an Event Related Spectral Pattern (ERSP) classifier using the saved data, and the example ERSP classifier training code
  - Build a continuous feedback testing block, with
    - Feedback display
    - Data signal processing and classifier application
  - Speed-up Matlab drawing by making a plot and drawing objects once, and thereafter setting properties on a handle to the drawn object

Today's Plan

- Review : ERP Viewer example. Solutions and discussion of problems
- Hands-on : Imagined Movement 1 – Calibration Block

break

- Hands-on : Imagined Movement 2 – Classifer training
- Hands-on : Imagined Movement 3 – Epoch Feedback

break

- Brain-test : Imagined Movement
- (Optional) Hands-on: Imagined Movement 4 – Continuous Feedback

Discussion : ERP Viewer

- Given the basic double-nested sequences and epochs table, adding annotate data, and process the selected data is (relatively) easy.
- Matlab drawing commands are **slow**, really **slow** ...
- Putting the stimuli and ERP computations in separate processes can help to avoid this (a bit) – if you try to re-draw every 1s and drawing takes 1.2s you still end up with laggy pictures..

Task : Imagined Movement

Experiment Task:

- Build a complete imagined movement based BCI experiment consisting of 3 blocks:
  1) Training/Calibration Block : where the user performs cued left and right hand movements
  2) Classifier Training Block : where the saved labelled data from the calibration block is used to train an ERSP classifier
  3) Testing Block : where the trained classifier is used to predict which hand the subject is imagining moving and this prediction is used to give the participant feedback about what the classifier though they were doing

Hands-on : Calibration Block

Experiment Task

- In 5 sequences of 5 epochs:
  - Display a 'baseline' cue for 1 seconds
  - Display a left/right cue, e.g. the strings L or R, for 3 seconds. Get the sequence to display from a block setting
  - Clear the display (or display a '+') and wait for 2 second (inter-epoch gap)
  - Move to the next epoch
- Display a 'Press key to continue string' between sequences, and wait for key press to move to the next sequence
- After the last sequence, display a thankyou message
- For every user cue, i.e. point when a L/R cue is displayed, record 3s of data annotated with the cue
- When the block is finished save the saved annotated training data

Solution hints

- As before 2 interacting MATLAB processes
  - imCalibrateStimulus_skel.m – calibration stimulus presentation
    - This is largely the same as the runStimulus from the ERP viewer, except for the addition of the baseline phase.
    - Remember to send start/finish events so signal processor knows when to stop storing data
  - imCalibrationSignals_skel.m – calibration block signal processing (essentially just data recording)
    - buffer_waitData can do most of the work for you!
    - Remember to remove the final exit event from the data-set!

Hands on : Classifier Training Block

Experiment Task

- Load the calibration data saved previously
- Pre-process and train an ERSP classifier
  - The frequency range of interest should be specified in the block-file
- Save the trained classifier for later

Start from:

- imTrainingSignals_skel.m – classifier training block signal processing (essentially just call to buffer_train_ersp_classifier)

Key functions

`clsfr=buffer_train_ersp_clsfr(data,devents,hdr,...)`

- train a linear classifier on the frequency power spectrum of the data
- `data, devents` – are data and associated events as output by buffer_waitdata.
- devents.type is used as the unique label for the class of data

Useful Options to change the signal-processing pipeline:

- capFile – cap file to use, e.g. 1010
- spatialfilter – type of reference to use, e.g. 'CAR','slap'
- freqband – frequency range used for classifier training
- badchrm – do we do bad channel removal?
- badtrrm – do we do bad trial removal?

Hands on : Epoch Feedback

Experiment Task

- Display some instructions for the user
- Show the epoch screen as for the calibration block, i.e. fixitation point, targets, baseline color change etc.
- At the end of the trial give the participant feedback about the classifiers prediction for example by displaying the strings, L or R.
- Use the trial data and trained classifier to make a prediction

Key functions

[*f*,*fraw*,*p*]=buffer_apply_ersp_clsfr(*X*,*clsfr*)

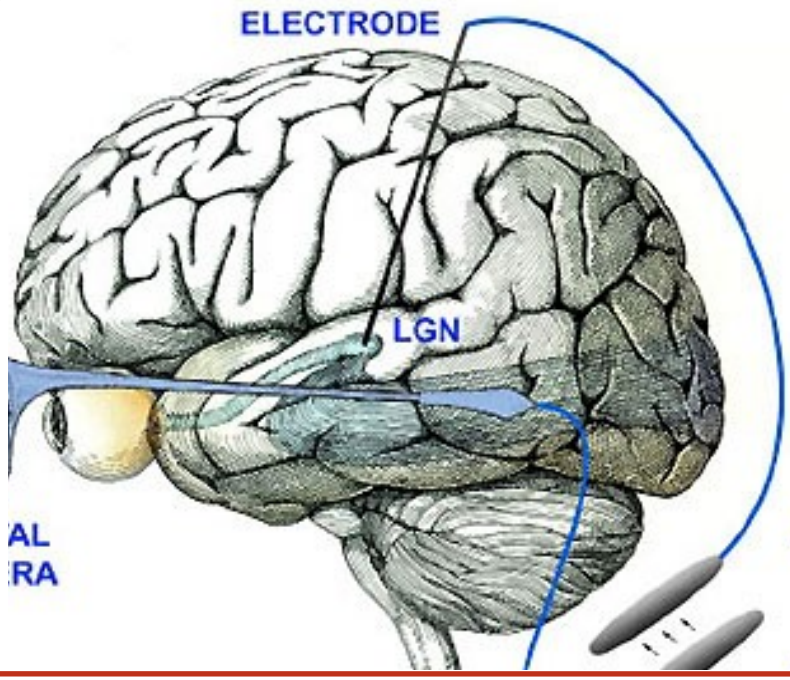
- Apply the ERSP pre-processing and trained classifier stored in *clsfr* to the input [channels x time] data in *X*
- *f* is the classifiers output decision value
  - decision value is a real number where f<0 predicts class -1, f>0 predicts class +1
  - combine decision values from different data by simply adding them, e.g. $f(X_a \& X_b) = f(X_a) + f(X_b)$
- *p*=*Pr*(+|*X*) is the estimated probability of the positive class

Solution hints

- As before 2 interacting MATLAB processes
  - imFeedbackStimulus_skel.m – feedback stimulus presentation
    - This is largely the same as for the calibration block, except:
      - No instruction of what type of movement to make
      - After the epoch has finished give visual feedback (target color to green) to indicate the the BCI has predicted they were doing
  - imFeedbackSignals_skel.m – feedback signal processing
    - Just apply the trained classifier to data after the epoch start event and generate a prediction event

Brain Test

- Test you system using a real participant
  - Hint: to make the biggest signals while testing use Actual Movements, and switch to imagined when the software is working correctly.



Hands on : Continuous Feedback

Experiment Task

- Display some instructions for the user
- Draw a **ball** at the center of the screen
- Continously process the data
  - Every 125ms get 250ms of data and apply the classifier to it
  - Accumulate all the classifier predictions
- Continuously update the feedback
  - Every 250ms compute the average of the last few seconds of classifier prediction probabilities
  - Use this average prediction to move the position of the ball where p=0 is at the left side, and p=1 is at the right side

Key functions : rectangle

- h=rectangle('position',[x,y,w,h],'curvature',1)
  - Draw a **circle** centered at x+w/2,y+h/2
- N.B. To move the rectangle you can use the returned handle, e.g.
  - set(h,'position',[x,y,w,h])
  - This is **much** faster than re-drawing the whole figure...

Solution hints

- As before 2 interacting MATLAB processes
  - imFeedbackStimulus_skel.m – feedback stimulus presentation
    - This is largely the same as for the calibration block, except:
      - No instruction of what type of movement to make
      - Ball moves based on classifier output to give the users feedback about what the BCI has detected
  - imFeedbackSignals_skel.m – feedback block signal processing
    - Remember to combine decision values before sending a prediction event

Summary

- Each experiment phase, (calibration, training, feedback) has it's own stimulus and signal-processing code
- ERSP classifiers can work with variable analysis lengths.
- Sum classifier decision values to combine information from different time points.