Brain-Computer Interface based on Event-Related Potentials of the EEG

NeurotechEU lecture, March 5, 2025

Michael Tangermann Jordy Thielen

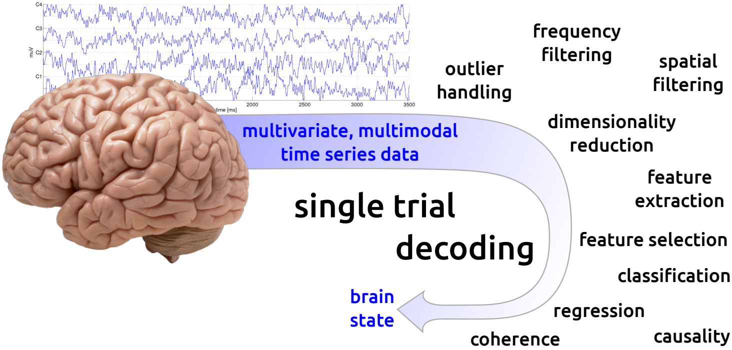
Radbound University

Lecture Overview

1. Brain-Computer Interfaces and Single-Trial Decoding of Brain Signals
2. How are ERP Responses Evoked?
3. An EEG Refresher
4. Event-Related Potentials: From EEG Channels to Features
5. The ERP Analysis Pipeline
6. Influence of Training Set Characteristics upon Model Quality

Lecture Overview

Machine Learning to Decode the Ongoing Brain State

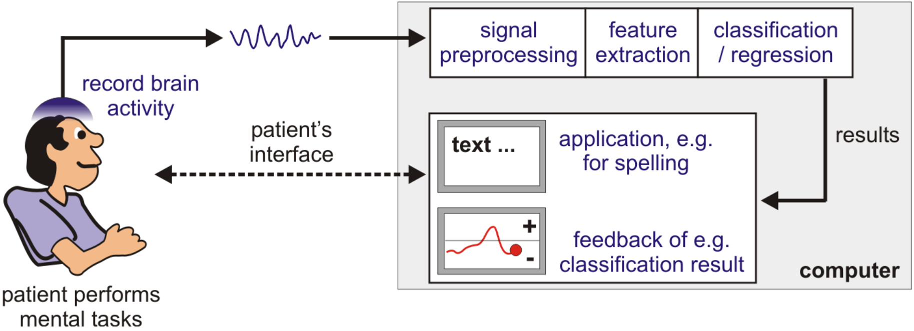


- Data: changing electric/magnetic fields: LFP, ECoG, (s)EEG, MEG
- Individual signals
- μV amplitudes but strong noise, contaminations by artifacts
- Limited datasets but medium-to-large dimensionality
- Typical: data is neither independent nor identically distributed.

Applications of Single Trial Decoding ...

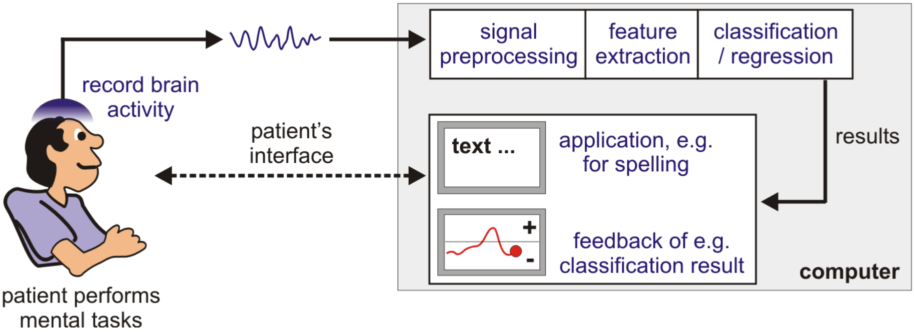
Which applications of single-trial decoding are you aware of? 

Brain-Computer Interface (BCI)



What tasks lead to **class-discriminative** EEG signals?

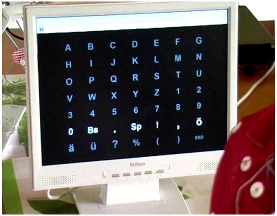
Brain-Computer Interface (BCI)



What tasks lead to **class-discriminative** EEG signals?

1. focus of attention to one of several external stimuli (focus of this lecture!)
2. self-initiated mental imagery tasks

BCI Example: Spelling Using Event-Related Potentials ($\rightarrow$ motor-impaired patients)



- Electroencephalogram (EEG) is measured and analyzed
- BCI user focusses attention/gaze on a symbol of the spelling matrix
- Visual enhancement of (random subsets, or rows and columns) elicit brain signal responses $\rightarrow$ (visual) event-related potentials (**ERP**)
- Classification model is trained to discriminate the responses of attended symbols from those of ignored symbols
- Graphical user interface allows to spell letter-by-letter

[Footage: BCI speller of Tübingen Group]

BCI Example: BCI-Supported Language Rehabilitation

Training for stroke patients with aphasia, using an auditory ERP protocol with word stimuli [Musso et al., Brain Communication, 2022]

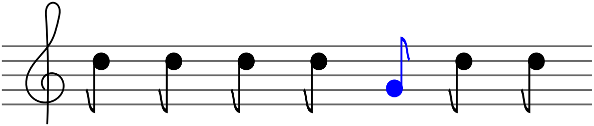
Video of online training

- Presentation of bisyllabic words in rapid sequence
- User task: attend appearances of target word, ignore non-target words
- Feedback by BCI supports patients to find individual strategy for rehabilitation

Lecture Overview

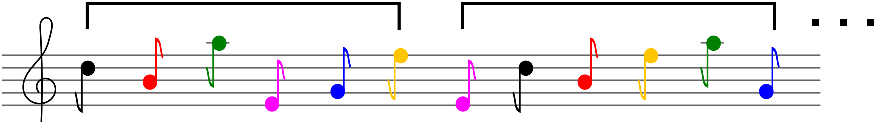
Reminder: Auditory Event-Related Potential (ERP) Protocols

- Late ERP components: use a classic **oddball** design
  - Different events/stimuli (in randomized order)
  - Frequent non-target / standard stimuli
  - Rare target / deviant stimuli



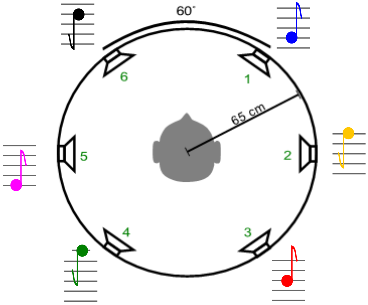
Reminder: Auditory Event-Related Potential (ERP) Protocols

- Late ERP components: use a classic **oddball** design
  - Different events/stimuli (in randomized order)
  - Frequent non-target / standard stimuli
  - Rare target / deviant stimuli



Reminder: Auditory Event-Related Potential (ERP) Protocols

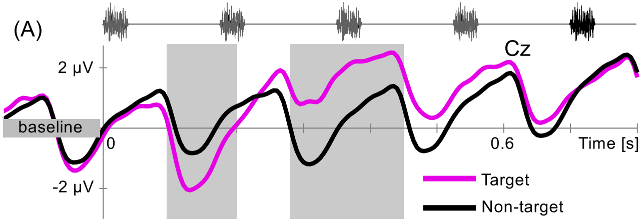
- Late ERP components: use a classic **oddball** design
  - Different events/stimuli (in randomized order)
  - Frequent non-target / standard stimuli
  - Rare target / deviant stimuli
  - Target stimulus can be defined by task instruction



Auditory Event-Related Potentials (ERP) of Target and Non-Target Stimuli

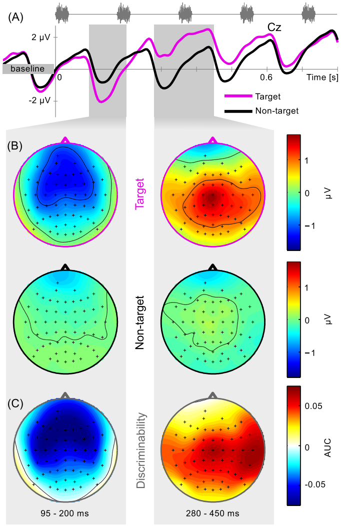
- Brisk tone stimuli (stimulus onset asynchrony (SOA) of 175 ms)
- Pseudo-randomized sequence of, e.g., 75 non-targets and 15 targets forms one *trial*.

An average *epoch* differs between target and non-target stimuli:



Attentive processing of a tone makes a difference!

Auditory Event-Related Potentials (ERP)
Feature Extraction



Classification of ERPs Evoked by Target vs. Non-Target Stimuli

- Discriminative ERPs (textbook):
  1. early negative component (N100-N200)
  2. late positive component (P300a/b)
- Features:
  1. discriminative intervals: average potential per interval and channel
  2. per-epoch covariance matrix → Riemannian geometry
  3. raw data
- Classification of target- vs. non-target epochs:
  1. regularized linear discriminant analysis
     [Hübner et al., Frontiers in Human Neurosci., 2018], [Sosulski et al., Neuroinformatics, 2020], [Sosulski et al., J Neural Eng., 2022]
  2. projection into tangent space + logistic regression
     [Kolkhorst et al., IEEE/RSJ IROS, 2018]
  3. deep learning with convolutional networks
     [Schirrmeister et al., Human Brain Mapping 2018]

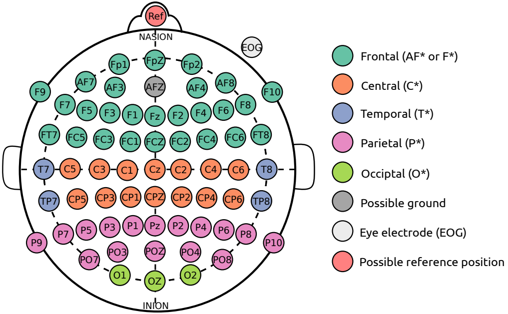
Lecture Overview

Make EEG results comparable between different labs

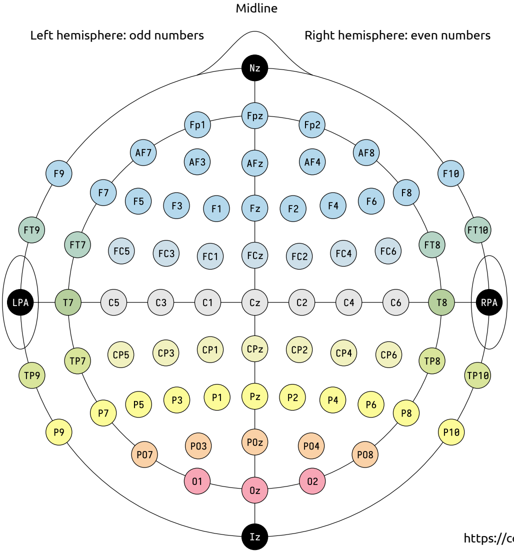
Problem: how can EEG laboratories compare / reproduce results, if they use different software, hardware or channel count?

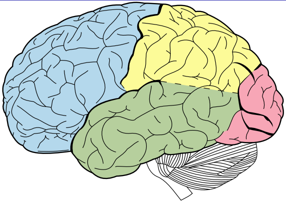
Problem: how can EEG laboratories compare / reproduce results, if they use different software, hardware or channel count?

→ precisely describe protocol (publish script), standardize recording procedures, EEG channel positions ("layout") and channel names.



International 10-10 system for EEG channels





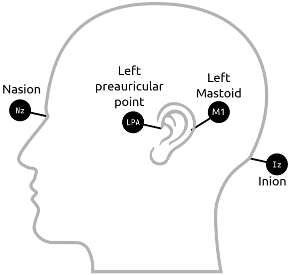


Image modified after Laurens R. Krol, CC0,
https://commons.wikimedia.org/w/index.php?curid=96859272

[Klem et al. (1999), Electroenc. and Clin. Neurophys. (Suppl.)]

International 10-10 system for EEG channels

- First letter indicates rough position over corresponding lobes / areas of the brain (see color code).
- Use letter combinations to add intermediate electrodes, e.g., FC row is between F row and C row. Exeption: AF between Fp and F row.
- Location of **channel Cz**: half distance between nasion (dip between eyes at upper end of nose) and inion (occipital ridge, bump), and between preauricular points (just in front of each ear)
- Midline electrodes Fpz, AFz, Fz, FCz, Cz represent 10 % steps from nasion (Nz) to Cz channel.
- Electrodes with numbers 1, 3, 5, 7, 9 for the left hemisphere represent 10 % steps of the LPA-to-Cz distance.
- Lower resolution: 10-20 system, higher resolution: 10-5 system ("F1", "FFC1", "FC1", "FCC1", "C1")

[Robert Oostenveld, Peter Praamstra (2001). Clin. Neurophys. 112 (4): 713–719.]

Types of EEG electrodes and systems

Important factors of electrodes:

- **gel** vs. **water** vs. **hydro-gel** vs. **dry** electrodes

Types of EEG electrodes and systems

Important factors of electrodes:

- **gel** vs. **water** vs. **hydro-gel** vs. **dry** electrodes
  - SNR and BCI performance higher for gel electrodes
    [Schwartz et al. (2020), Front. Neurosci.]
  - usability, user acceptance?
- passive electrodes vs. active, i.e., pre-amplified electrodes
- shielded vs. non-shielded cables / electrode bodies

Types of EEG electrodes and systems

Important characteristics of amplifiers / systems:

- noise level (good systems have approx. 1μV standard deviation)
- resolution steps (16 bit or 24 bit analog-to-digital converter?)
- signal range (saturation / clipping problems?)
- susceptibility for external noise / artifacts?
- form factor, channel count, portability / **mobility**, cost

Lecture Overview

Visualizing ERPs: single responses → averaged response

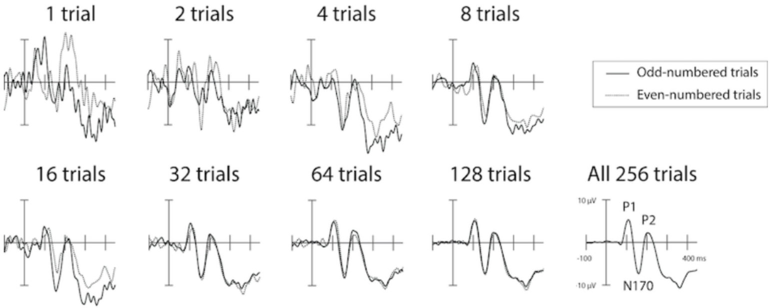


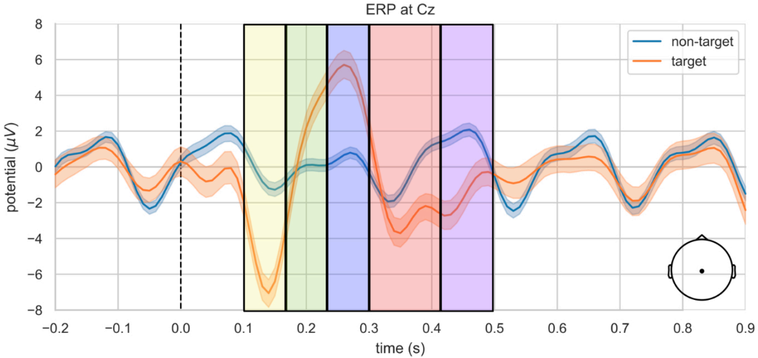
Fig. 12 Effects of averaging increasing numbers of trials in an ERP experiment, for a single human participant. Stimuli were images of human faces. There were a total of 256 trials in the experiment, and the average of these is shown at the bottom, with prominent component peaks (see next section) labelled. #

[https://neuraldatascience.io by Aaron J Newman, CCA-NC-SA 4.0)]

- Average ERPs of odd and even ~~trials~~ epochs should look identical.
- Due to noise, evoked potentials emerge only after averaging.

Note: In non-BCI ERP research, often *negative axis up* is used, and ERP response are not overlapping due to long inter-stimulus intervals.

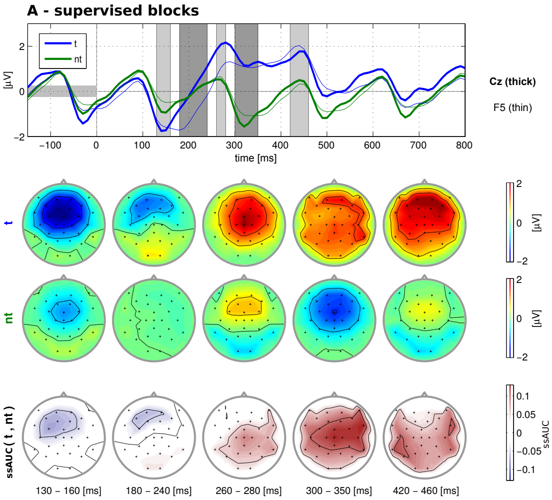
ERP-BCI example:
channel Cz, visual evoked potential (VEP) response



- At time t=0, either a target stimulus is presented ("target event"), or a non-target stimulus ("non-target event").
- Averaging epochs of the same class, we observe systematic differences in the ERP / VEP responses of target and non-target stimuli.

Why do we look at time intervals? 🖐🖐

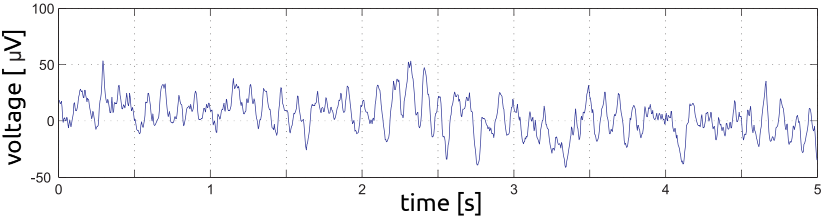
Example: auditory ERP responses (AEP), scalp patterns



What was the time between stimulus onsets (i.e., stimulus onset asynchrony, "SOA") in this recording?

From time series to features (part I)

A time series signal of a single EEG channel consists of *samples*:

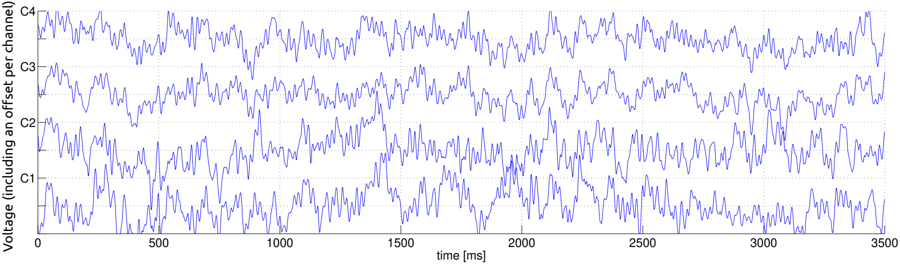


During supervised calibration of a BCI, we measure raw time series signals, and enrich them with additional meta information:

- (large, binary) file containing voltage values of EEG measurements. Size: *number of channels* × *number of samples*
- information about the sampling rate and channels ("workspace"): layout, channel names, impedances, recording date etc.
- marker information: *What happened when during the recording?* time stamps and type of corresponding **events**

From time series to features (part II)

Four EEG channels: C1, C2, C4, C4

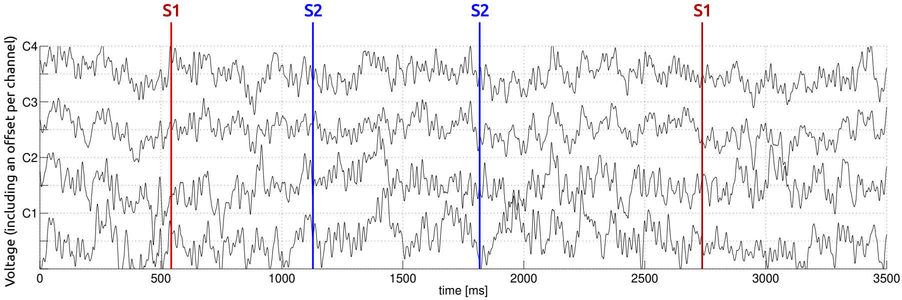


For every event / epoch / stimulus that happens during the recording, do:

- extract some features per channel
- concatenate features of all channels into a D-dimensional feature vector $\mathbf{x}$ (i.e., one data point)

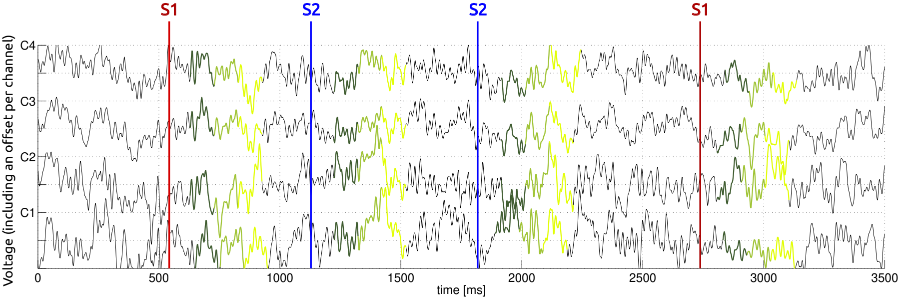
Finally, stack all feature vectors, i.e. data points, into a data matrix $\mathbf{X}$; Events types will define the class labels $\mathbf{y}$. We are ready to train an LDA!

Example: N events, 4 channels, 3 temporal features/chan.



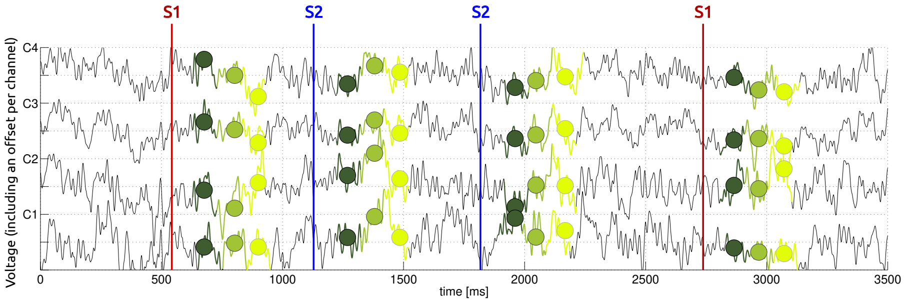
- Events (→ data points) of either target (S1) or non-target (S2) class
- Extract one epoch (time window) per event, typically overlapping

Example: N events, 4 channels, 3 temporal features/chan.



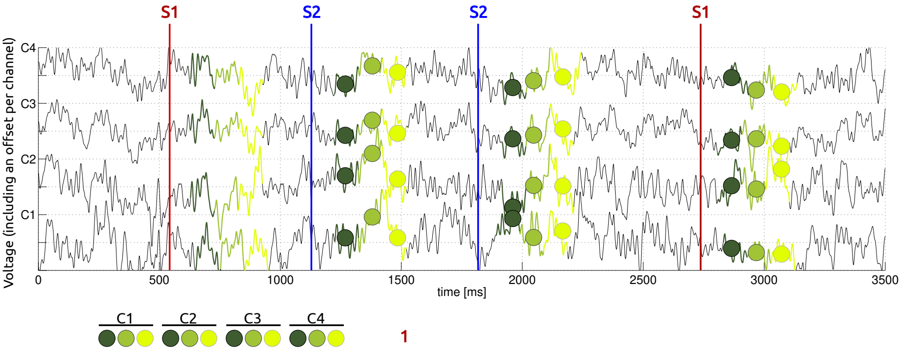
- Events (→ data points) of either target (S1) or non-target (S2) class
- Extract one epoch (time window) per event, typically overlapping
- Characterize data point $\mathbf{x}_i$ by 3 features per channel, concatenate!

Example: N events, 4 channels, 3 temporal features/chan.



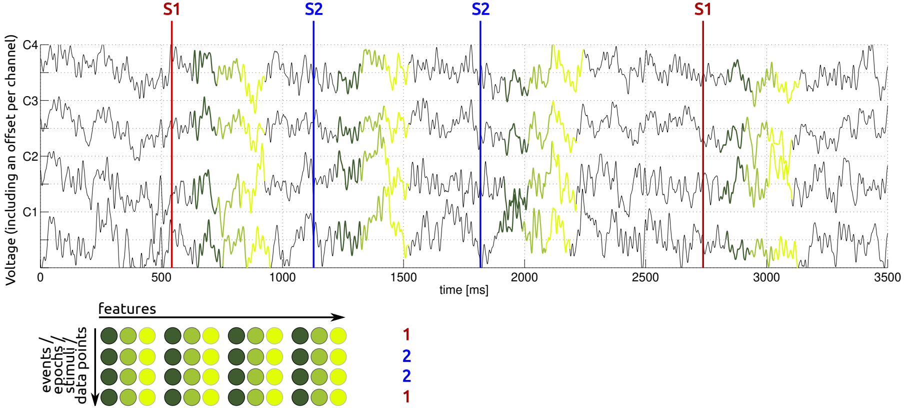
- Events ($\rightarrow$ data points) of either target (S1) or non-target (S2) class
- Extract one epoch (time window) per event, typically overlapping
- Characterize data point $\mathbf{x}_i$ by 3 features per channel, concatenate!

Example: N events, 4 channels, 3 temporal features/chan.



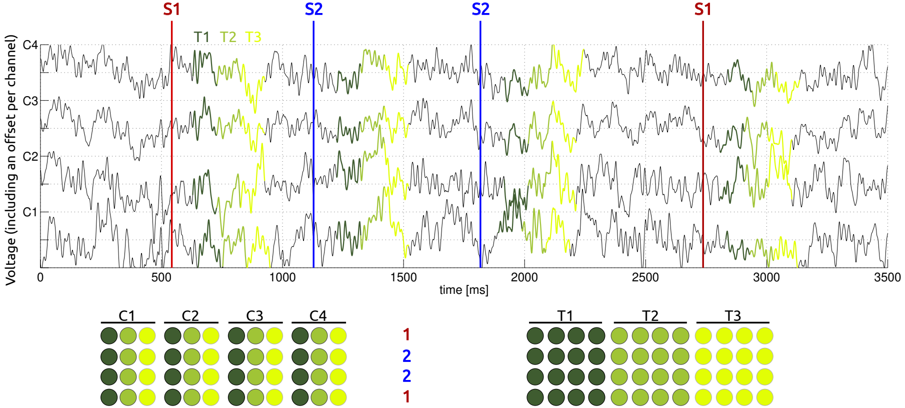
- Events ($\rightarrow$ data points) of either target (S1) or non-target (S2) class
- Extract one epoch (time window) per event, typically overlapping
- Characterize data point $\mathbf{x}_i$ by 3 features per channel, concatenate!

Example: N events, 4 channels, 3 temporal features/chan.



- Events ($\rightarrow$ data points) of either target (S1) or non-target (S2) class
- Extract one epoch (time window) per event, typically overlapping
- Characterize data point $\mathbf{x}_i$ by 3 features per channel, concatenate!
- Store in data matrix $\mathbf{X} \in \mathbb{R}^{N \times D}$: one row per event/data point, each containing $D = 3 * 4 = 12$ features (num_feat $*$ num_chan).

Example: N events, 4 channels, 3 temporal features/chan.

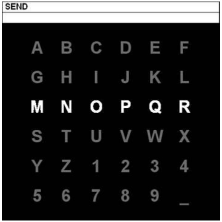


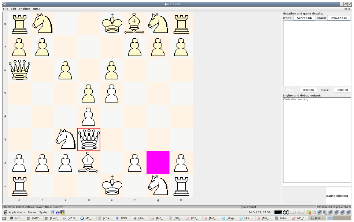
- Events (→ data points) of either target (S1) or non-target (S2) class
- Extract one epoch (time window) per event, typically overlapping
- Characterize data point $\mathbf{x}_i$ by 3 features per channel, concatenate!
- Store in data matrix $\mathbf{X} \in \mathbb{R}^{N \times D}$: one row per event/data point, each containing $D = 3 * 4 = 12$ features (num_feat $*$ num_chan).

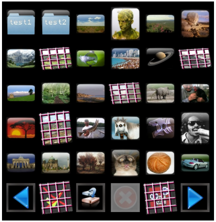
Lecture Overview

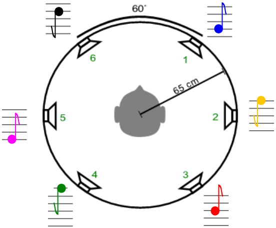
Typical ERP analysis pipeline

Goal: train/calibrate a decoding pipeline first, then use it for an online ERP-BCI application.









Typical ERP analysis pipeline

Part 1: Load training (calibration) data and preprocess it

1. Load time series data, event markers and workspace information
2. (Optional, for performance reasons and for ICA): Limit frequency content (Nyquist! Aliasing!) by a bandpass filter, e.g., to $[0.5, 45]$ Hz, then downsample to 100 Hz.
3. Inspect and remove outlier channels:
   - **High variance**: Bad impedance, high noise
   - **Low variance**: Dead electrode? Signal in saturation (clipping)?
   - **Diff**: reveals sudden strong jumps. Was electrode touched? Lost contact, bad gelling, moving dry electrode?
4. (Optional, good for later visualization, not necessarily improves classification): Inspect and remove artefact subspaces via ICA
   - Train PCA+ICA to derive independent components
   - Inspect and remove artifact components (eye blinks, eye movements, muscle artifacts, pulse artifacts, ...)
   - Project back to channel space (for ERP visualization) or continue working in ICA space (full rank!)

Typical ERP analysis pipeline

Part 1: Loading and preprocessing (cont.)

5. Windowing of continuous data into **epochs** of a few hundred milliseconds. Each epoch should comprise the evoked potential(s) after a stimulus.
6. Inspect and **mark outlier epochs** for later removal:
   - threshold for variance
   - threshold for min-max difference
   - delete the epochs data structure, but remember the position / index of outlier epochs
7. To avoid filter artifacts, **start over with continuous data**: filter to low frequency band, e.g., [0.5, 16] Hz. Idea: everything outside of an ERP's frequency range is removed.

Typical ERP analysis pipeline

Part 2: Epoching and feature extraction

8. Windowing of continuous data into epochs
9. Remove epochs marked as outliers (see step 6)
10. (Optional, benefit depends on dataset and decoding models):
    For every channel and epoch: correct for drift / offsets using a baseline interval, e.g., [-SOA ms to 0 ms] relative to stimulus.
11. For every channel, extract the average potential of a few time intervals, e.g., in steps of 50 ms for early ERPs, longer intervals for P300 and later ERPs.
12. arrange features and labels into training dataset $(\mathbf{X}, \mathbf{y})$

Generally: store all preprocessing parameters, thresholds etc., as they may need to be applied in the online application, too.

Typical ERP analysis pipeline

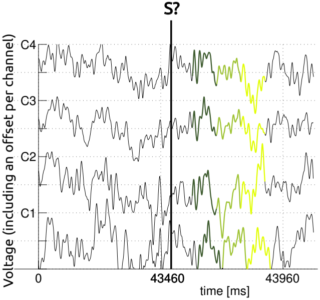
Part 3: Training the decoding model (classifier!)

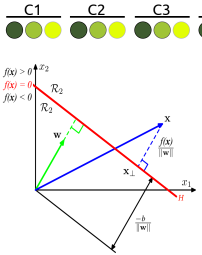
13. Train either LDA or logistic regression classifier (or any other classification model of your liking)
    - If hyperparameters need to be determined (e.g. regularization strength, learning rate, channel subset etc.), then either use nested chronological crossvalidation on training data or a training- and validation split.
14. Estimate generalization error using chronological crossvalidation or holdout/test set.
15. If you are happy, train one model on the full data. Store the model for online use.

Typical ERP analysis pipeline

Part 4: Online application

16. Record ongoing brain signals, obtain small data packets with low latency from hardware recording system.
17. Preprocess according to Part 1 (possibly NOT rejecting outlier epochs?)
    - frequency filtering: store filter state after each data packet
18. Once all packets of an epoch have arrived: extract features according to Part 2
19. Classify the features with the model trained in step 15.

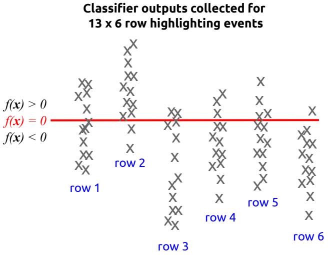




Typical ERP analysis pipeline

Part 4: Online application

16. Record ongoing brain signals, obtain small data packets with low latency from hardware recording system.
17. Preprocess according to Part 1 (possibly NOT rejecting outlier epochs?)
    - frequency filtering: store filter state after each data packet
18. Once all packets of an epoch have arrived: extract features according to Part 2
19. Classify the features with the model trained in step 15.
20. For all epochs belonging to one trial: collect & post-process the classifier outputs to determine the target symbol:



Role of bias?

Typical ERP analysis pipeline

Part 4: Online application

16. Record ongoing brain signals, obtain small data packets with low latency from hardware recording system.
17. Preprocess according to Part 1 (possibly NOT rejecting outlier epochs?)
    - frequency filtering: store filter state after each data packet
18. Once all packets of an epoch have arrived: extract features according to Part 2
19. Classify the features with the model trained in step 15.
20. For all epochs belonging to one trial: collect & post-process the classifier outputs to determine the target symbol:

    Role of bias? 🖐🏾🖐🏽
21. Execute the resulting control step of that trial (i.e., write a symbol) in the final application and provide feedback to user.

Typical ERP analysis pipeline

Part 4: Online application

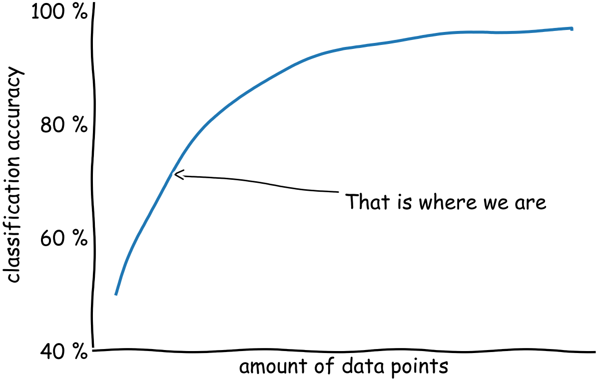
16. Record ongoing brain signals, obtain small data packets with low latency from hardware recording system.
17. Preprocess according to Part 1 (possibly NOT rejecting outlier epochs?)
    - frequency filtering: store filter state after each data packet
18. Once all packets of an epoch have arrived: extract features according to Part 2
19. Classify the features with the model trained in step 15.
20. For all epochs belonging to one trial: collect & post-process the classifier outputs to determine the target symbol:

    Role of bias?
21. Execute the resulting control step of that trial (i.e., write a symbol) in the final application and provide feedback to user.

22. (Optional): include dynamic stopping of stimulus presentation to shorten a trial.
23. (Optional): include online adaptation of the classifier to cope with changing feature distributions.

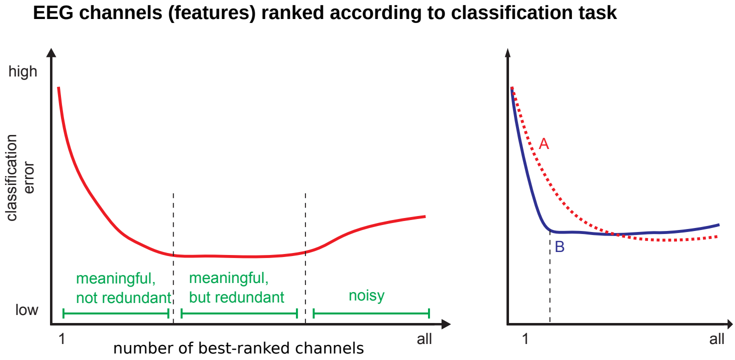
Lecture Overview

Training set characteristics → classifier performance



- More training data → asymptotically better performance (if data is stationary).

Training set characteristics → classifier performance



- More training data → asymptotically better performance (if data is stationary).
- More channels (features) *typically* lead to better performance, if lots of training data is available.
- If training data is limited, try reducing the channel set (according to domain knowledge, priors, or via channel selection method).

Wrap-Up

Having heard this lecture,
you . . .

- know what an ERP is and know protocols that elicit ERPs.
- can explain the nomenclature of electrodes names.
- can explain how multi-channel continuous voltage recordings are mapped to form a feature vector and training data.
- have an understanding of how training set size and number of features influence the quality of the decoding model
- can design an ERP decoding pipeline, exploring the influence of spatial vs. temporal vs. spatio-temporal features. The practical assignment and data can be downloaded here: https://github.com/thijor/eeg_tutorial_erp