1. Lecture 4.1 Adaptivity in BCIs
   1. Unknown Parameters
      1. For most BCI questions and implementations, the parameters leading to best accuracy (W, b,…) are *a priori* unknown…
         1. Depend on highly variable factors
            1. Sensor placement
            2. Subject state
            3. Etc…
         2. Different for every person, task, montage, etc.
         3. Depend on hard-to-measure factors
            1. E.g., brain functional map
         4. Depend on expensive-to-measure factors
            1. E.g., brain folding
      2. How to solve this problem?
   2. Model Calibration
      1. Can use calibration / training data to estimate parameters from, and a separate calibration step
      2. Calibration data -> calibration step -> BCI Model
         1. BCI model is like spatial filters
   3. Prior Knowledge
      1. Prior knowledge is neuroscientific, such as:
         1. Anatomical atlases
            1. E.g. Talairach, LONI
            2. “This part of function is primarily found in this part of the brain”
         2. Functional Atlases
            1. If available
            2. Most are probalistic
         3. Timing information
            1. E.g. neural latencies, reaction times
         4. Frequency bands of oscillatory processes
            1. Alpha, beta, theta, …
   4. Calibration Data
      1. Example/calibration data is used to calculate optimal parameters of a BCI, and is extremely important
   5. The Ideal Calibration Data
      1. Collected with the same/similar measurement apparatus as used for online runs
         1. Otherwise extra transformations and uncertainty incurred
         2. If you use EEG to calibrate, use EEG to record
      2. Comprises multiple independent realizations / repetitions / trials (to quantify variability)
         1. One-shot learning (one exemplar) is much harder
         2. Much easier to learn and fit a model from multiple trials
      3. Collected under conditions that are as close to those of the online runs as possible (i.e., drawn from the same statistical distribution)
         1. Same person is preferable
         2. Same sensor arrangement is preferable
         3. Same session is preferable
         4. Task parameters (stress level, …) should be similar
      4. Obviously a cost/benefit tradeoff:
         1. Would trade off some performace for being able to reuse one recording for multiple sessions and persons
      5. If there is a bias (e.g. different session), data should cover multiple realizations (e.g., multiple sessions) to capture variability
      6. A plain EEG recording is “unlabeled” (no knowledge about the association between raw observed signal and the cognitive state variable of interest)
      7. Labeled data (person is “surprised” / “not surprised”) is far more useful than unlabeled
         1. Much much more useful
      8. Labels are assigned per realization (e.g., per trial) and index the output that the BCI shall produce for this class of data
   6. Summary
      1. The required data to calibrate a BCI resembles data produced by controller psychological experiments
      2. Features
         1. Continuous EEG (or other)
         2. Multiple trials/blocks (capturing variation)
         3. Randomized (eliminating confounds)
         4. Even markers to encode cognitive state conditions of interest, e.g. stimuli/responses (called “target markers” in BCILAB)
      3. Can also be used for offline performance tests
2. Lecture 4.2 Machine Learning
   1. Machine Learning (ML)
      1. Large field with 100s of algorithms
      2. Most methods conform to a common framework of a training function and a prediction function
         1. Neural networks and fuzzy logic to name a few
      3. Machine learning method
         1. Two functions
            1. Training Function

Gets data and labels

Outputs Model

* + - * 1. Prediction Function

Get new data and model from training function

Outputs labels

* + 1. Intermediate model parameters θ capture the learned relationship
    2. Data and Labels / target values
       1. N is the number of trials
       2. F is the number of features
       3. D is the number of output dimensions
       4. Data and Labels are both matrices
    3. The Machine leanring framework is largely trial-based (learning from exemplars)
    4. Most methods come in form of two functions
       1. Learning function
          1. Usually 10 times more complex
       2. Prediction function
    5. Learning function is often *far* more complex than the prediction function
  1. Sub-Types in ML
     1. Supervised Learning
        1. Given a set of (input, output) pairs as training data, learn a parametric (or “Non-parametric”) model M that encodes the mapping from input to output
        2. The most applicable to BCI because we have the EEG and the outcome we want
     2. Unsupervised Learning
        1. Given a set of training examples, learn the structure in the input space
           1. E.g. clusters, manifolds, probability density
     3. Semi-Supervised Learning
        1. Some training examples have labels, others do not
        2. If labels are sometimes hard to obtain
     4. Others
        1. E.g. Active Learning, Online learning, …
  2. Related Areas
     1. Probability Theory
     2. Statistics
        1. Because your samples are part of a much larger population
     3. Optimization
        1. The parameters that you come up with are supposed to be optimal given the data and such
     4. Neural Networks
     5. Artificial Intelligence
  3. Using Machine Learning
     1. Often, one trial segment (sample) is extracted for every target marker in the calibration recording and is used as training exemplar
     2. Its associated label can be deduced from the target marker
     3. What is considered optimal depends on extra assumptions (a.k.a. priors)
  4. Detour: Feature Extraction
     1. Caveat: Off-the-shelf machine learning methods often do not work very well when applied to raw signal segments of the calibration recording
        1. Too high-dimensional (too many parameters to fit)
        2. Too complex structure to be captured (too much modeling freedom)
     2. Solution: Introduce additional mapping (called “Feature extraction”) from raw signal segments onto feature vectors
        1. Output is often of lower dimensionality
        2. Hopefully statistically “better” distributed (easier to handle for machine learning)
        3. Identify key features that you think are important such as
           1. I think its spectral perspective and I think it’s only at 10 Hz and I think phase is irrelevant so let’s throw that away effectively reducing the complexity and dimensionality of the entire mapping.
  5. Examples of Feature Extraction
     1. Depends on the process of interest (e.g. oscillation, ERP peak, complex phenomenon)
     2. For oscillations, e.g.:
        1. log-Variance (logarithm yields more convenient data distributions)
        2. Part of the Fourier Spectrum
     3. For ERPs, e.g.:
        1. Peak latency, height, width (artificial example)
        2. Mean in one or more time ranges relative to an event
        3. Subset of Wavelet coefficients

1. Lecture 4.3 Concrete Case Study
   1. Example Calibration Problem
      1. Task: A person is presented with a sequence of 300 images (one every 2 seconds). Half of the images are exciting, the other half are not. One channel of EEG (at Cz location) is recorded.
      2. Question: How to design a BCI that can determine whether a person is shown an exciting or a non-exciting image?
      3. Approach: For each trial k, cut out an epoch of of 1s length, extract a short vector of features , and assign a label in . Use machine learning to find an optimal statistical mapping from onto .
         1. E = excited
         2. NE = not excited
   2. Extracting Features of a Peak
      1. A supposed characteristic peak in a time window (relative to an event) could be characterized by three parameters:
         1. Width
         2. Latency
         3. Height
            1. The amplitude of the peak
         4. Every time a image is shown then we have these three numbers
   3. Resulting Feature Space
      1. Plotting the 3-element feature vectors for all exciting trials in red, and non-exciting trials in green, we obtain two distributions in a 3d space
         1. This is called the feature space, can get all the way up to 10,000 features.
   4. ML with Feature Extraction
      1. Including the feature extraction, the analysis process is as follows:
         1. EEG recording
         2. Multiple events happening that we annotate
         3. We cut out the chunk of information around each event and write down the label what the stimulus was.
         4. We extract the features
            1. i.e. calculate peak
         5. This create a feature vector for each event from our training set that we call matrix X,y
         6. Send to training function
         7. Get the model
   5. Using Machine Leaning
      1. The feature vectors are passed on to a machine learning function
         1. E.g. Linear Discriminant Analysis
            1. Determines a parametric predictive mapping
            2. Outputs a hyperplane and we can tell if it is on one side of the plane it is class one and on the other side it is class two

our two states “E” and “NE”

* + - * 1. Now we have a new observation now online that can be used
        2. Hyperplane is a vector that we call θ which encodes the orientation of the plane in the space, it has as many numbers in it as this space has dimensions.
  1. LDA In a Nutshell
     1. Given trial segments (in vector form) in and ,  
        ,   
        ,
        1. This is with equal E & NE
        2. Σ is the covariance matrix
  2. Resulting Predictive Map
     1. LDA Generates parameters of a linear mapping  
        1. Produces the line
     2. For classification, the mapping is actually non-linear  
        1. Takes the sign of output and outputs a 1 or -1
  3. More on LDA
     1. Assumptions
        1. Data in each class is distributed according to a Gaussian Distribution
        2. Shape of the distribution is identical for all classes
     2. Benefits
        1. Simple, fast and optimal in the large-sample limit if assumptions are true
     3. Problems
        1. Very sensitive to outliers (non-Gaussian)
        2. Covariance matrix estimates become unreliable / unusable for too few trials and too many dimensions
  4. Putting it together
     1. Apply band-pass filter to calibrated recording
     2. Extract epochs relative to target markers
     3. Extract features for each epoch (here: peaks)
     4. Submit all feature vectors & target labels to LDA to calculate ***θ*** and *b* of the predictive mapping
     5. For online operation, the overall prediction function consists of the band-pass filter, epoch extraction, then the feature extraction followed by the predictive map
     6. This yields a primitive “excitement detector”

1. BCI Models
   1. BCIs are described by “BCI models” that specify both the *filter graph* and the *prediction function* (including its parameters)
   2. These models are the result of the calibration step
2. BCI Paradigms
   1. BCI paradigms are a notion that was first developed in the BCILAB framework
   2. A BCI paradigm is the *full description and codification of a particular type of calibration and prediction process*
      1. Calibration Recording(s)
      2. Calibrate function (FUNCTION 1)
         1. Including signal processing
         2. This produces the BCI Model
      3. BCI Model (FUNCTION 2)