New Tools for Brain-Computer Interface Design

Christian A. Kothe

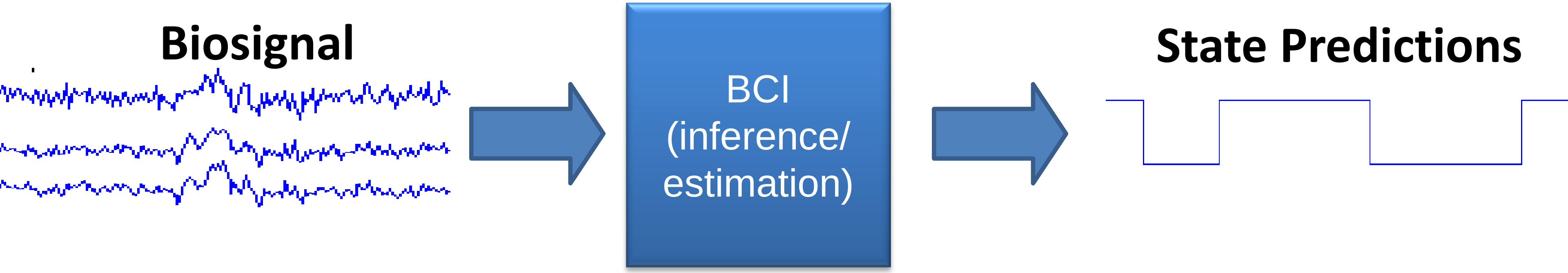
Swartz Center for Computational Neuroscience, UCSD

Outline

- Background
  - What is a BCI
  - What is BCILAB
- Theory
  - Overview
  - ERP approaches
  - Oscillatory approaches
- Practice
  - Toolbox overview
  - GUI & scripts walkthrough

What is a BCI/BMI?

- “A system which takes a biosignal measured from a person and predicts (in real time / on a single-trial basis) some abstract aspect of the person's cognitive state.”
  - Abstract aspect of cognitive state: “*type of limb movement imagined*”, “*degree of surprisal*”, “*type of vowel imagined*
  - Biosignal: EEG, ECoG, MEG, … (+ possibly non-brain data)



Research Directions

Research Directions

- **Clinical**: Communication and control devices for the severely disabled

Research Directions

- **Clinical**: Communication and control devices for the severely disabled
- **HCI**: User-state monitoring, intelligent assistive systems

Research Directions

- **Clinical**: Communication and control devices for the severely disabled
- **HCI**: User-state monitoring, intelligent assistive systems
- **Entertainment**: Computer game controllers

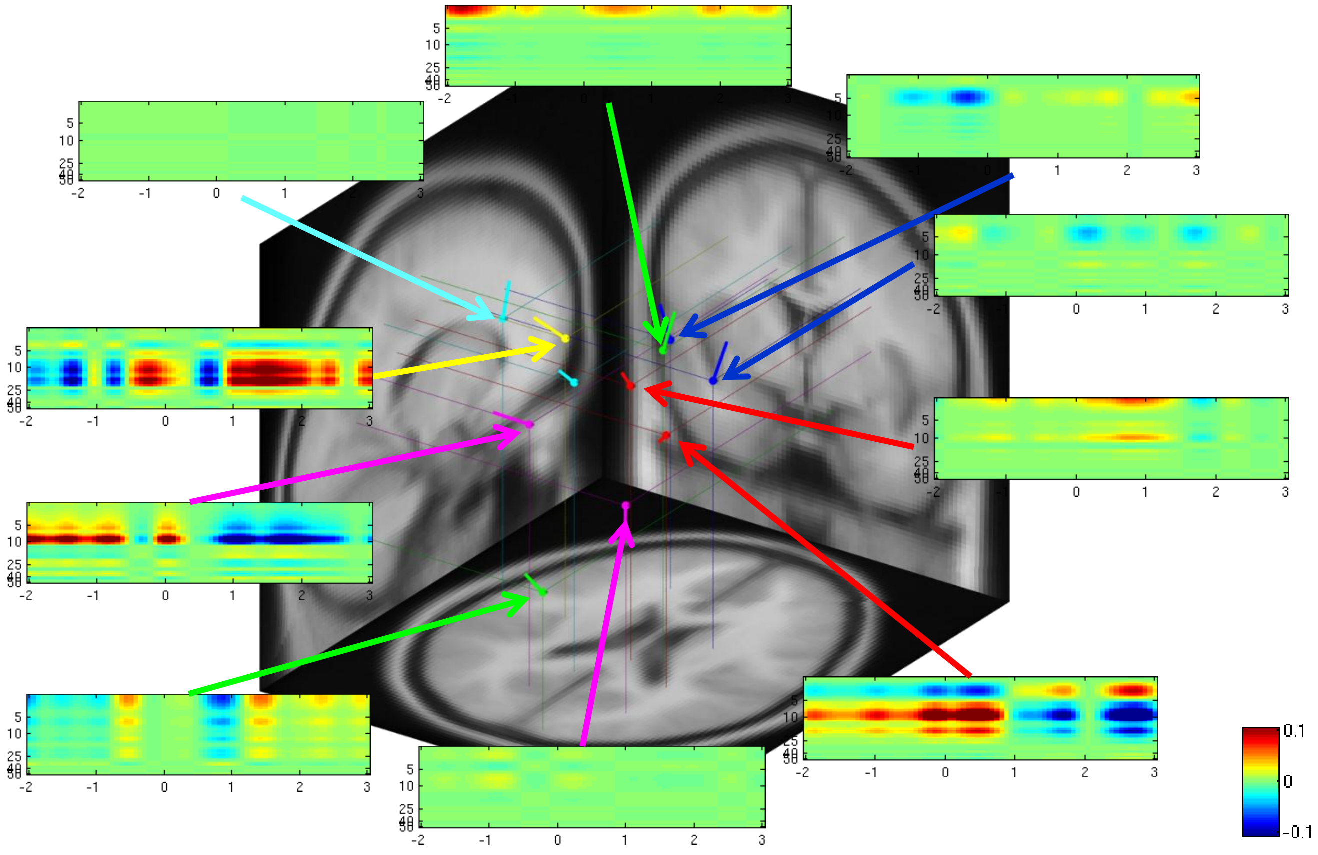
Swartz Center for Computational Neuroscience

Research Directions

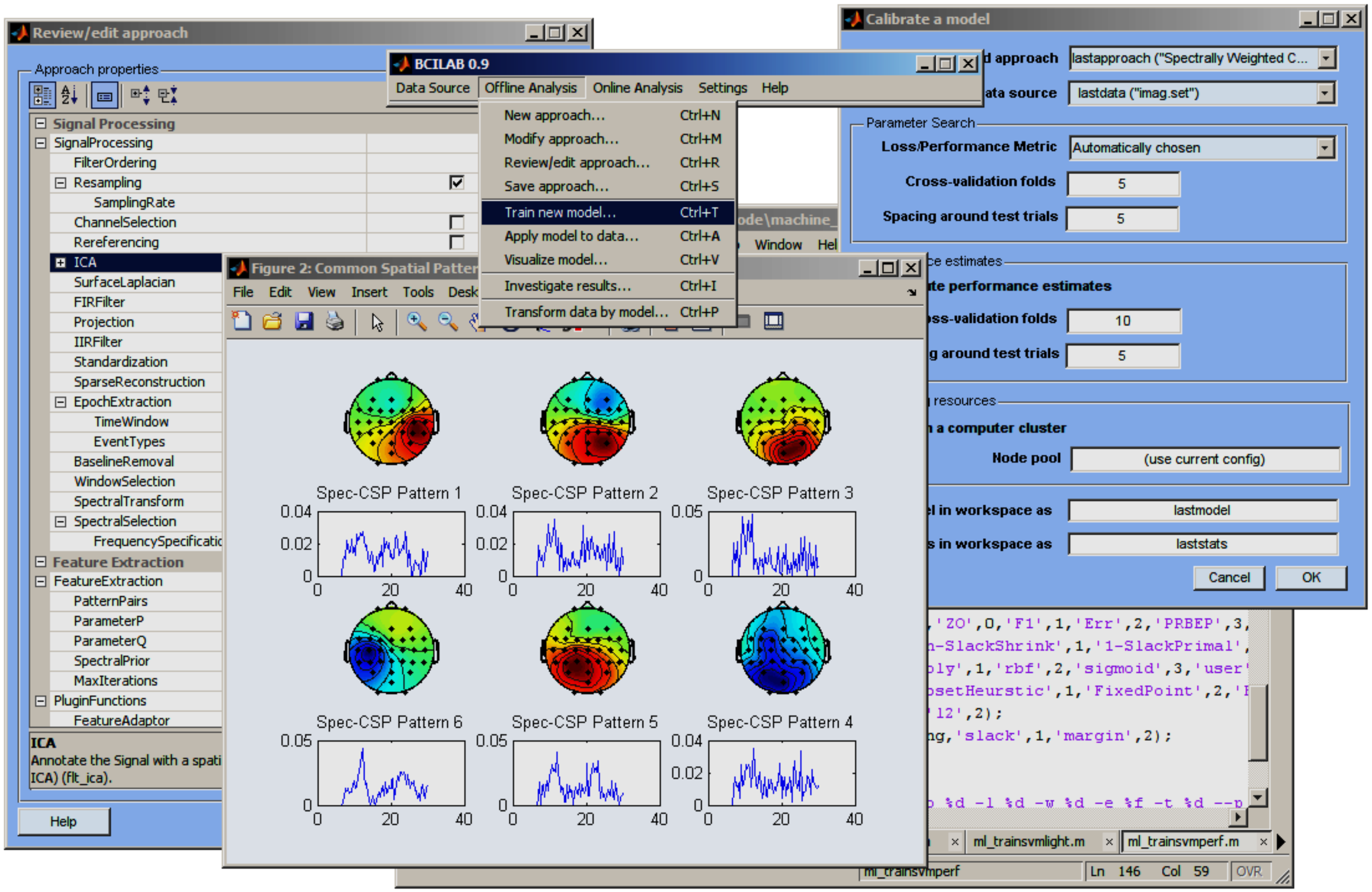
- **Clinical**: Communication and control devices for the severely disabled
- **HCI**: User-state monitoring, intelligent assistive systems
- **Entertainment**: Computer game controllers
- **Neuroscience**: Brain feedback experiments

Research Directions

- **Neuroscience**: also, *decoding models* of brain dynamics (exploratory research)

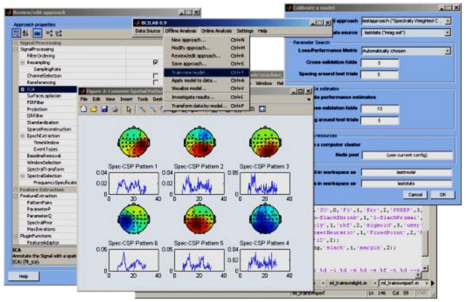


BCILAB



[http://sccn.ucsd.edu/**wiki/BCILAB**](http://sccn.ucsd.edu/wiki/BCILAB)

Summary



- Software environment for:
  - Design & *rapid* prototyping of cognitive state assessment (CSA) systems, both traditional and unconstrained approaches
  - Empirical performance assessment (offline/online)
  - Real-time use, prototype deployment
  - Large-scale batch analysis

BCILAB Specialty

- State of the art
- Largest collection of machine learning & signal processing components in any open-source BCI package
  - Many standard components (CSP, LDA, SVM, …)
  - Many modern components (SBL, SSA, AMICA, HKL, DPGMM, LR-DAL, …)
  - Some novel components (OSR, RSSD, SSB, …)
- Modern framework
  - Fully probabilistic
  - Model inference from data corpora
  - Neuroscience-informed features (e.g., anatomical priors)
  - Processing of parallel streams (MoBI)

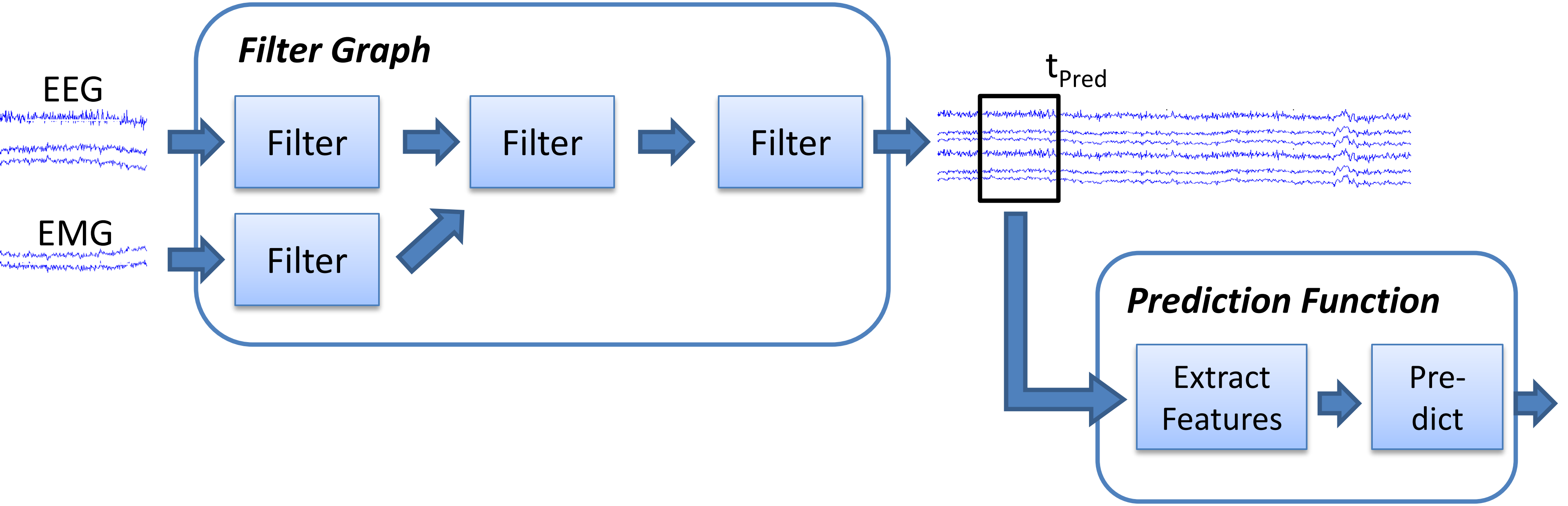
Long-Term Goals

- Probe landscape of possible approaches for real-world CSA & assess future performance limits
  - Replicating and re-purposing established BCI methods
  - Exploring larger-scale data, computation and complexity than usual
  - Leveraging neuroscience knowledge and infrastructure
  - Focusing on unified and principled methods where possible

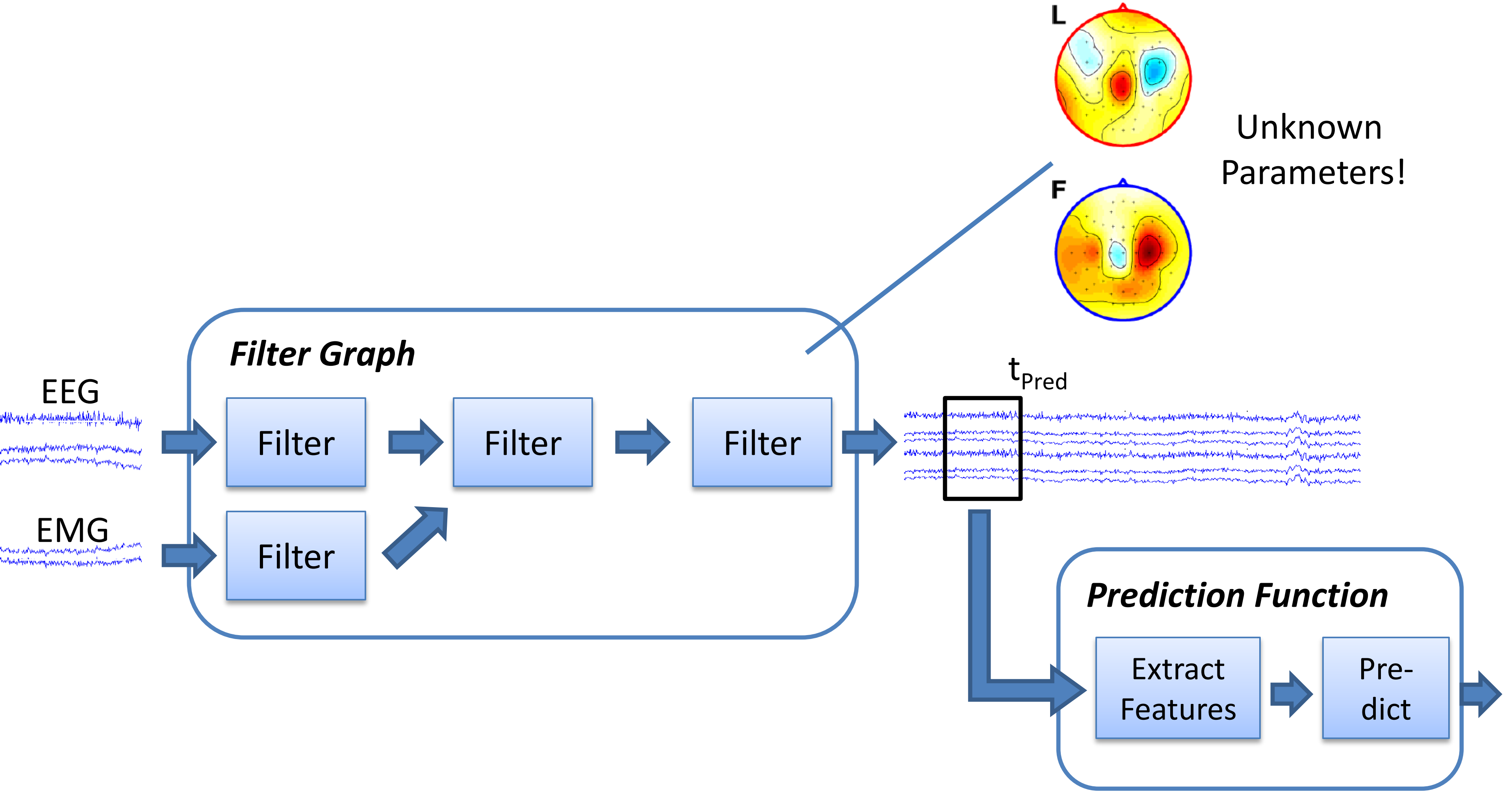
Outline

- Background
  - What is a BCI
  - What is BCILAB
- Theory
  - Overview
  - ERP approaches
  - Oscillatory approaches
- Practice
  - Toolbox overview
  - GUI & scripts walkthrough

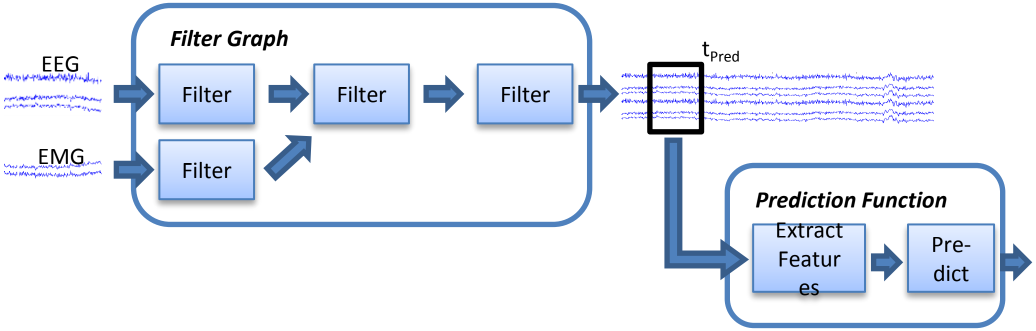
Information Flow



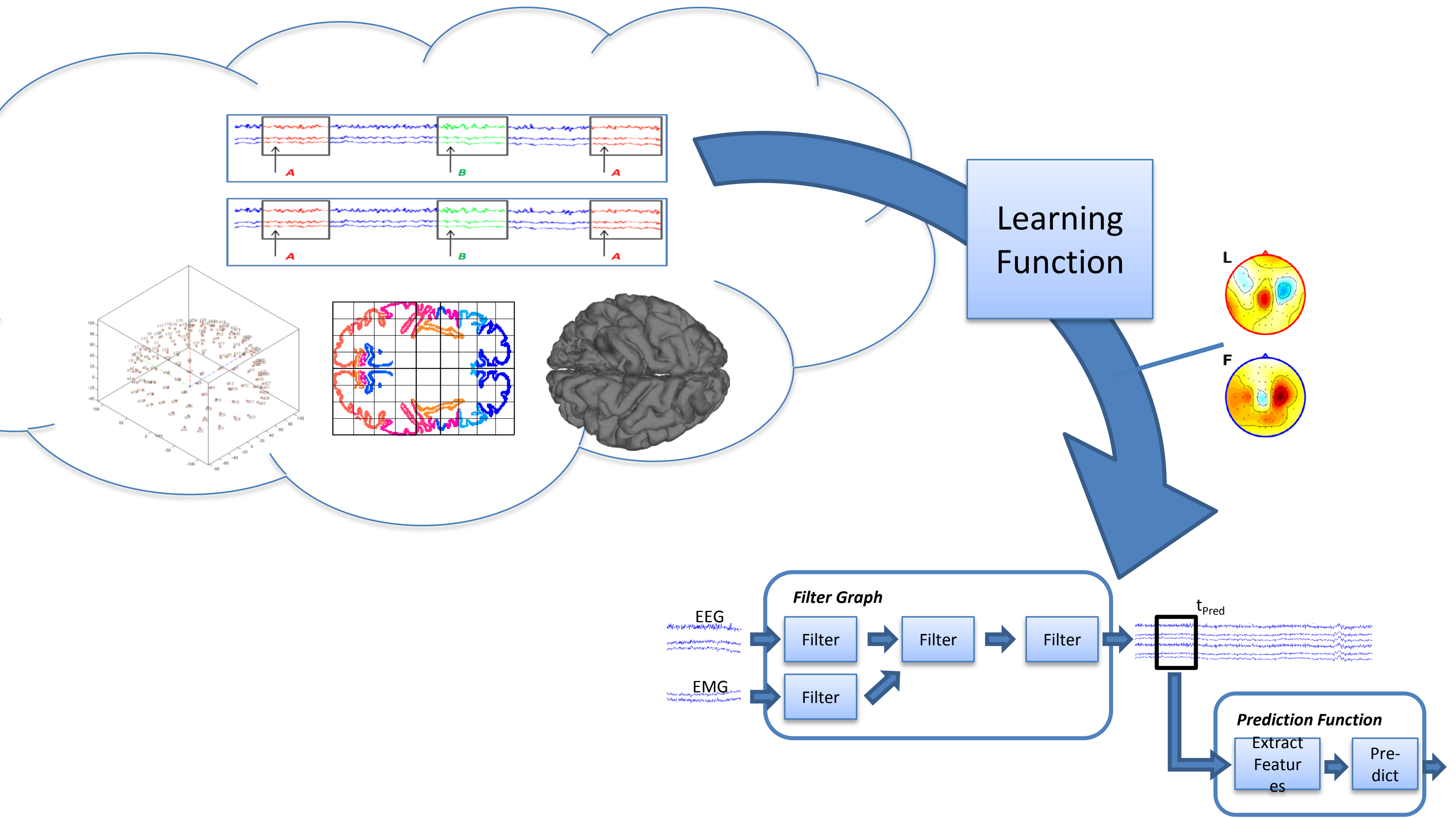
Information Flow



Information Flow

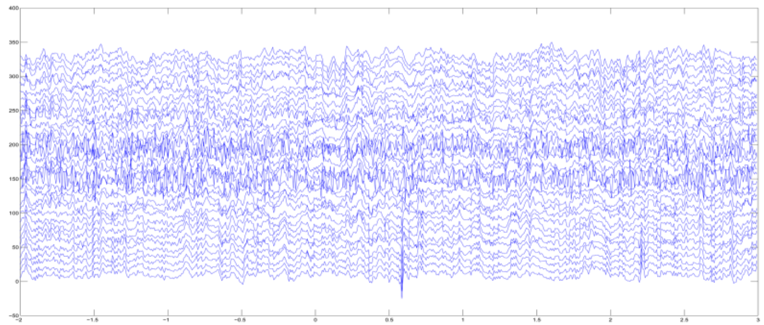


Information Flow



The Prediction Function

- Mathematical mapping

y = f(**X**); **X**=  y= "left hand" (-1)
"right hand" (+1)

- Functional form

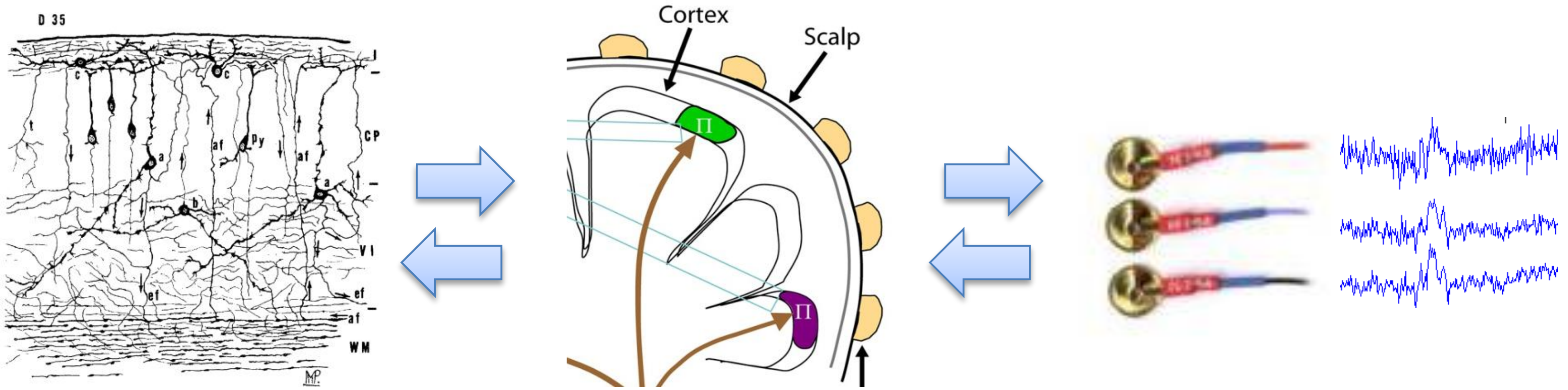
e.g., $y = \text{sign}(\boldsymbol{\theta}\text{var}(\boldsymbol{WX}) + b)$

- Unknown parameters

e.g., **W**, b, …

Functional Form

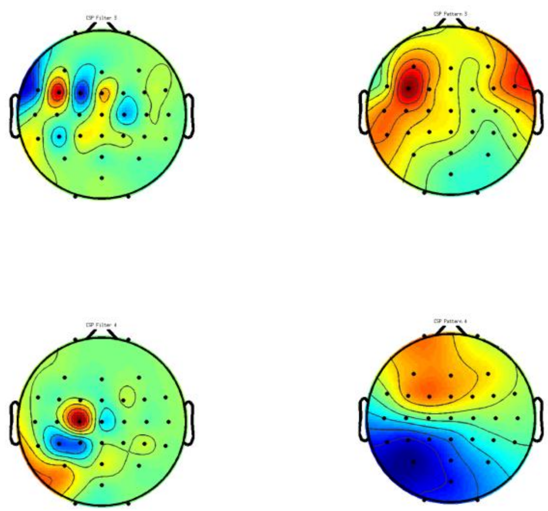
- Reflects the relationship between observation (data segment X) and desired output (cognitive state parameter y)
- Based on some assumed generative mechanism (forward model) or ad hoc



- Note: Functional form is the inverse mapping!

First Ingredient: Spatial Filter

- Linear inverse of volume conduction effect
  $\boldsymbol{X} = \boldsymbol{AS}$ (forward)
  $\boldsymbol{S} = \boldsymbol{WX}$ (inverse)
- Two example filters and forward projections:



W **A**

Further Ingredients

- Inverse mapping from source time courses to latent cognitive state, e.g.:

$$y = \boldsymbol{\theta}\,\mathbf{vec}(\boldsymbol{W}\boldsymbol{X}) + b$$ **(linear)**

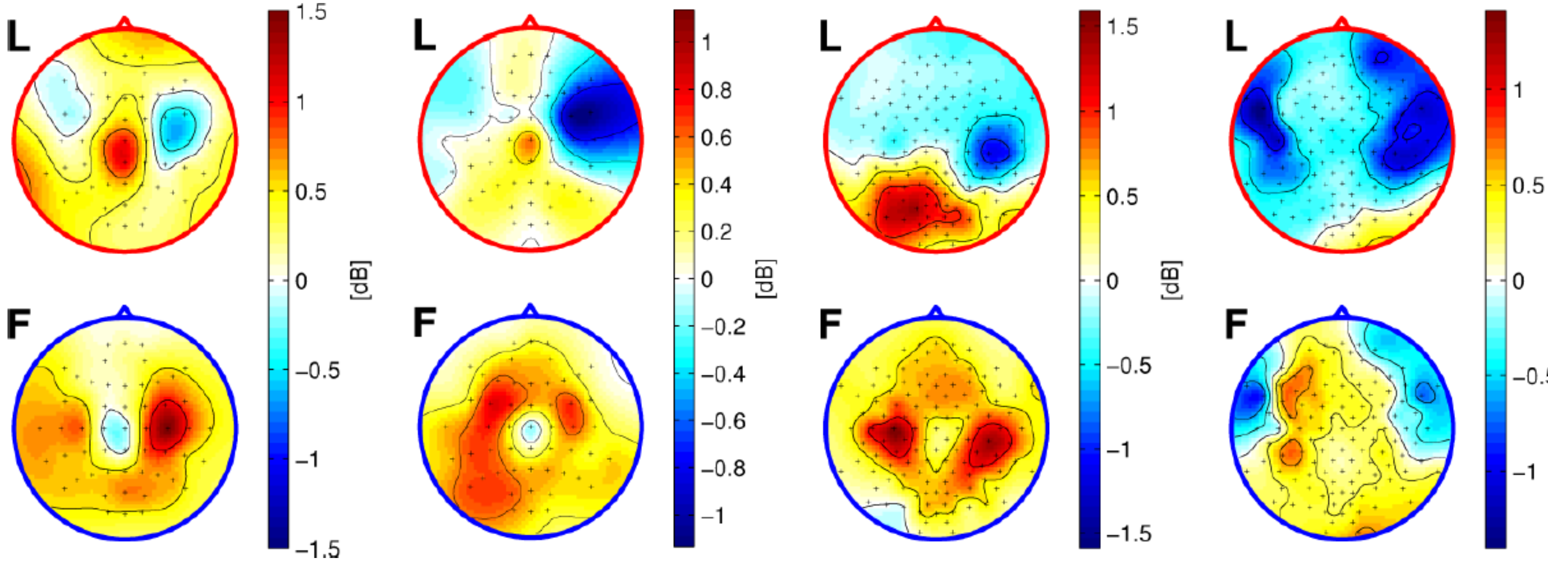
$$y = \boldsymbol{\theta}\,\mathbf{vec}(|(\boldsymbol{W}\boldsymbol{X})\boldsymbol{T}|) + b$$ **(nonlinear…)**

Unknown Parameters...

- for most BCI questions and implementations, the parameters leading to best accuracy (**W**,b, ...) are *a priori* unknown
  - Depend on hard-to-measure factors (e.g., brain functional map)
  - Depend on expensive-to-measure factors (e.g., brain folding)
  - Depend on highly variable factors (e.g., sensor placement, subject state)
  - Different for every person, task, montage, etc.

Unknown Parameters…

- Example per-channel parameters across four subjects:

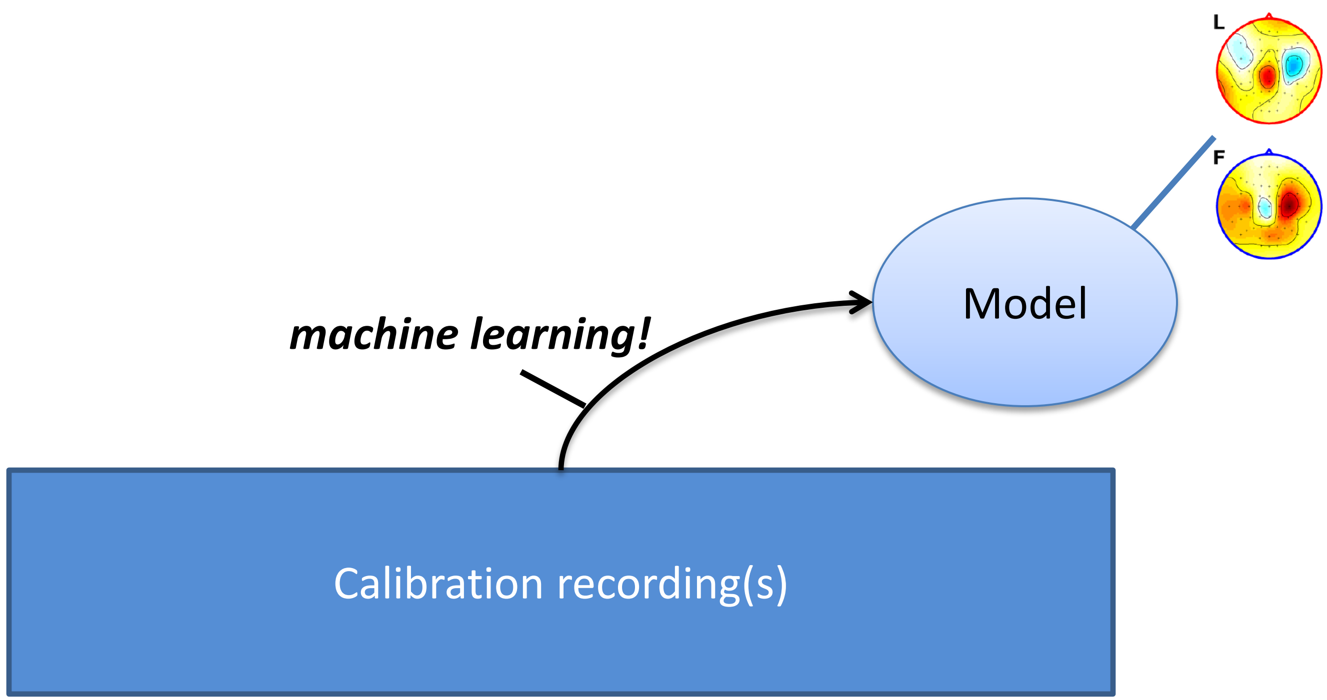


Person 1 Person 2 Person 3 Person 4

(image: Blankertz et al. 2007)

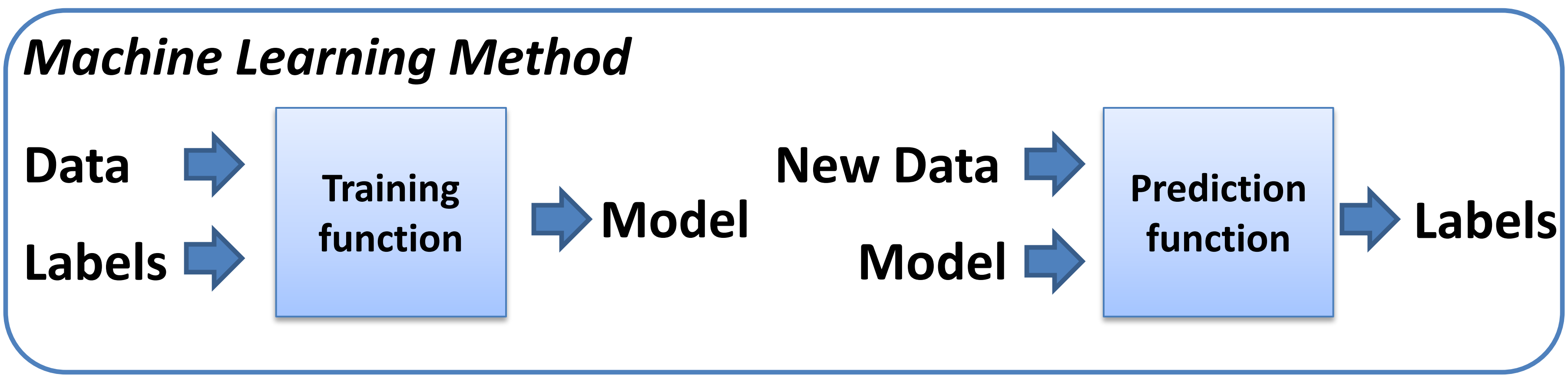
Model Calibration Today

- Modern standard approach: utilize data where both the BCI input (e.g. EEG) and desired output (cognitive state) is known and adapt BCI parameters using *machine learning* techniques



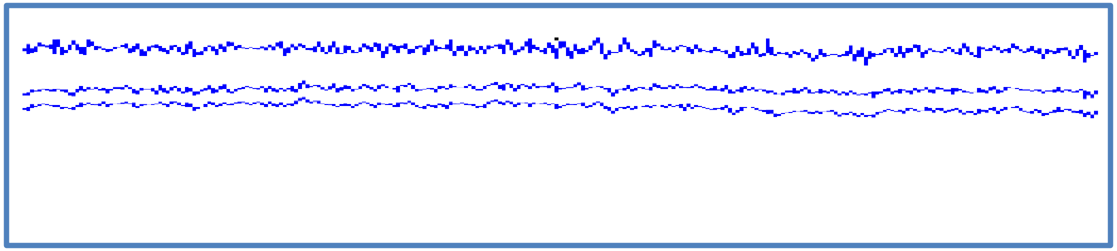
Machine Learning Refresher

- Large field with 100s of algorithms
- Most methods conform to a common framework of a *training function* and a *prediction function*
- Model parameters $\boldsymbol{\theta}$ capture the learned relationship
- Data $\boldsymbol{X} \in \mathbb{R}^{N \times F}$ and Labels / target values $\boldsymbol{y} \in \mathbb{R}^{N \times D}$
  N = #trials, F = #features, D = #output dims.



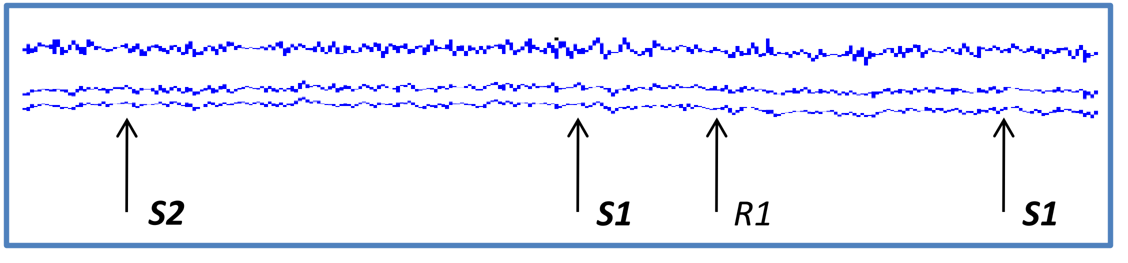
Desired Calibration Recording

- Standard psychological experiment
  - continuous EEG (or other)
  - multiple trials/blocks (capturing variation)
  - randomized (eliminating confounds)



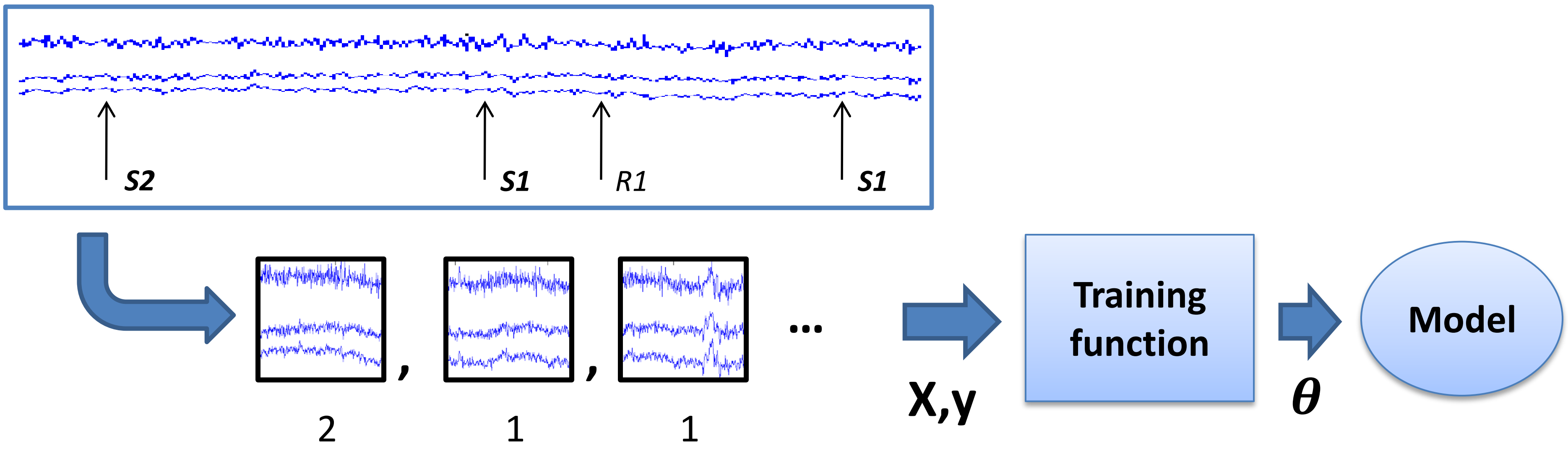
Desired Calibration Recording

- Standard psychological experiment
  - continuous EEG (or other)
  - multiple trials/blocks (capturing variation)
  - randomized (eliminating confounds)
  - often *event markers* to encode timing and type of cognitive state conditions of interest, e.g., stimuli/responses ("*target markers*" in BCILAB)



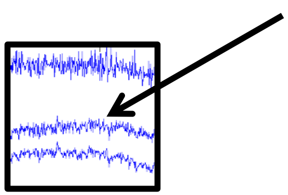
Using Machine Learning

- Often, one trial segment (sample) is extracted for every target marker in the calibration recording (length depends on timing of related phenomena)



Detour: Feature Extraction

- **Caveat:** Off-the-shelf machine learning methods often do not work very well when applied to raw signal segments of the calibration recording
  – too high-dimensional (too many parameters to fit)
  – too complex structure to be captured (too much modeling freedom)

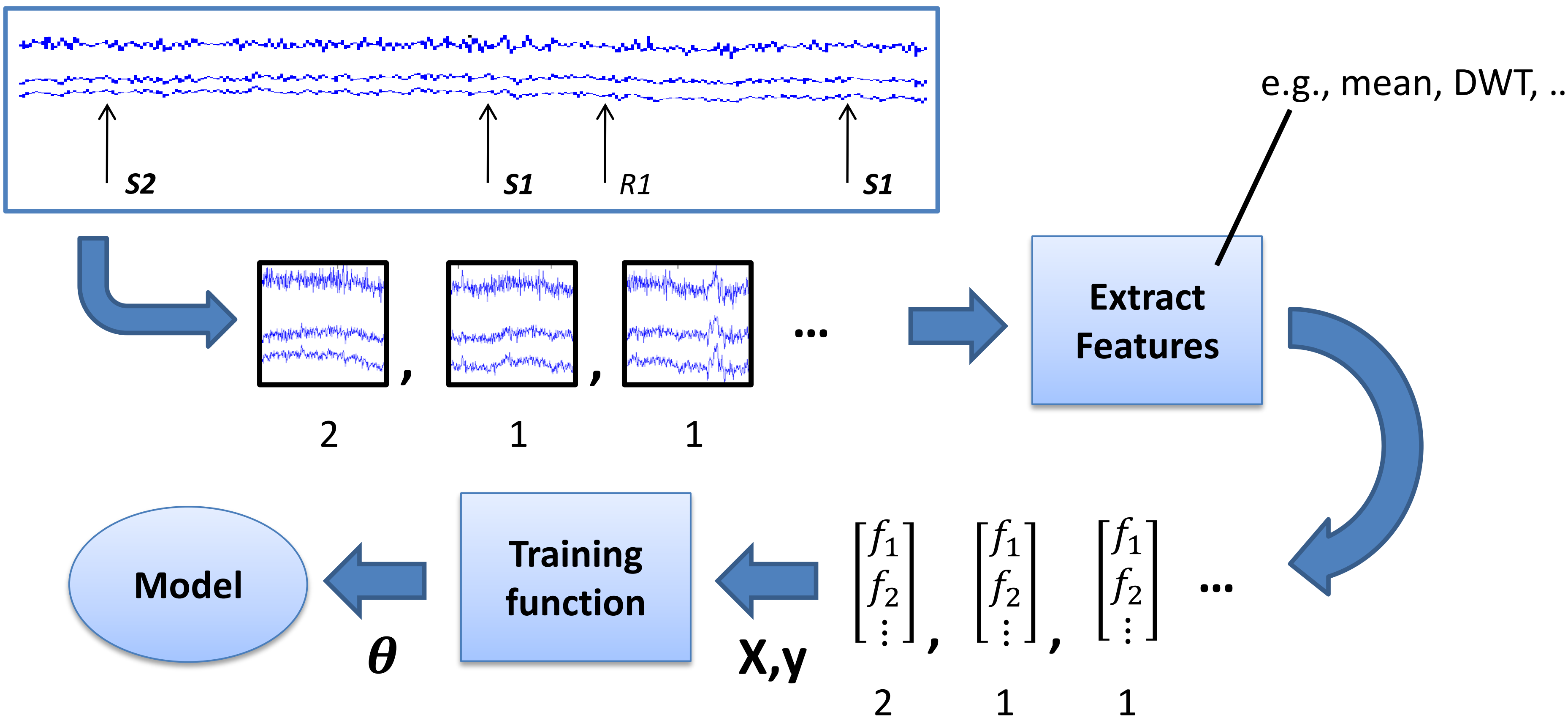


Detour: Feature Extraction

- **Solution**: Introduce additional mapping (called *"feature extraction")* from raw signal segments onto feature vectors
  - output is often of lower dimensionality
  - hopefully statistically "better" distributed (easier to handle for machine learning)

ML with Feature Extraction

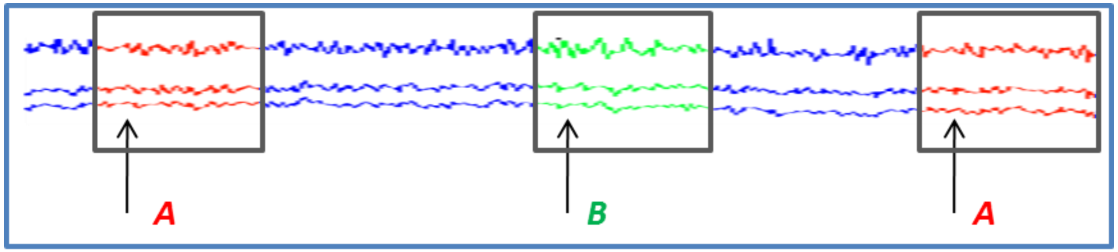
- Including feature extraction, the analysis process is as follows:



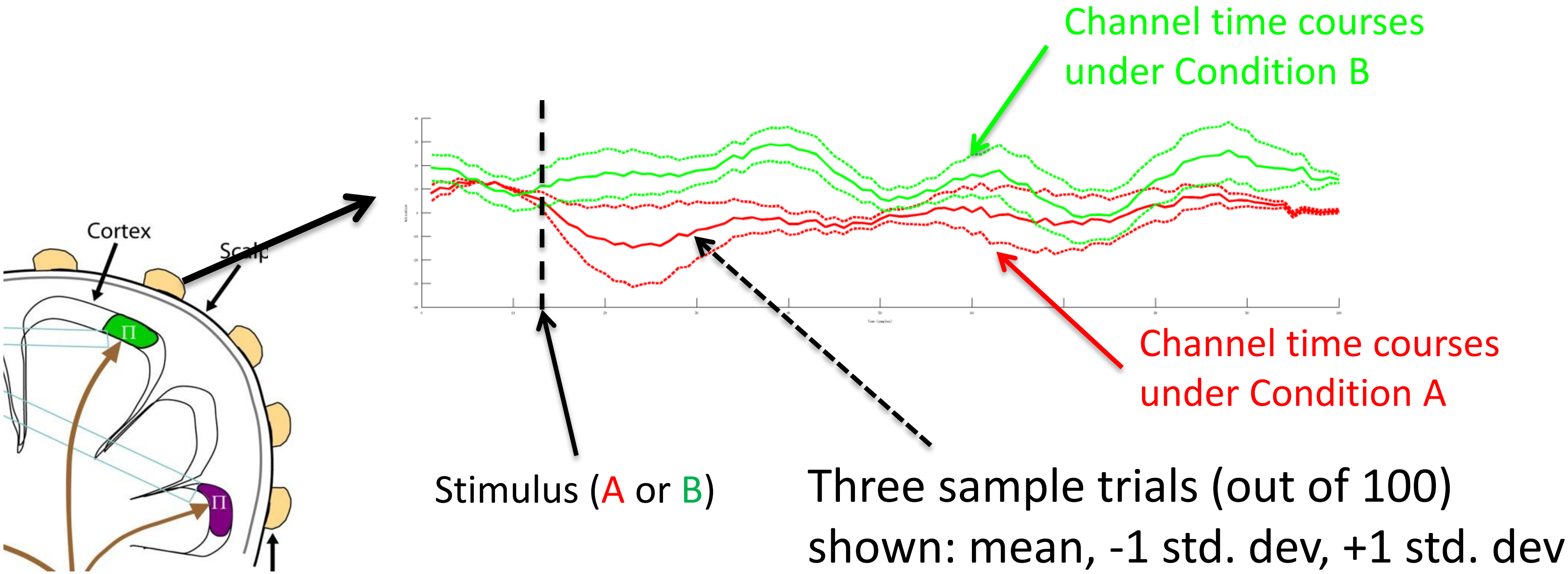
Two Major BCI Analysis Pathways

1. Simple Case: ERP-like Patterns

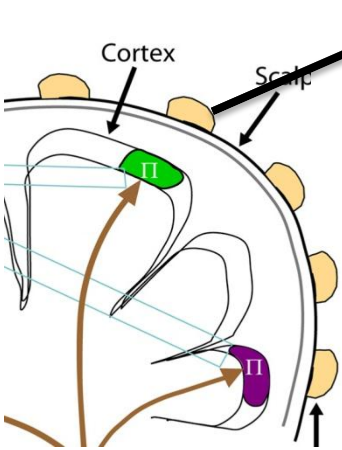
- Suppose a calibration recording with 100 stimuli of type A and 100 stimuli of type B

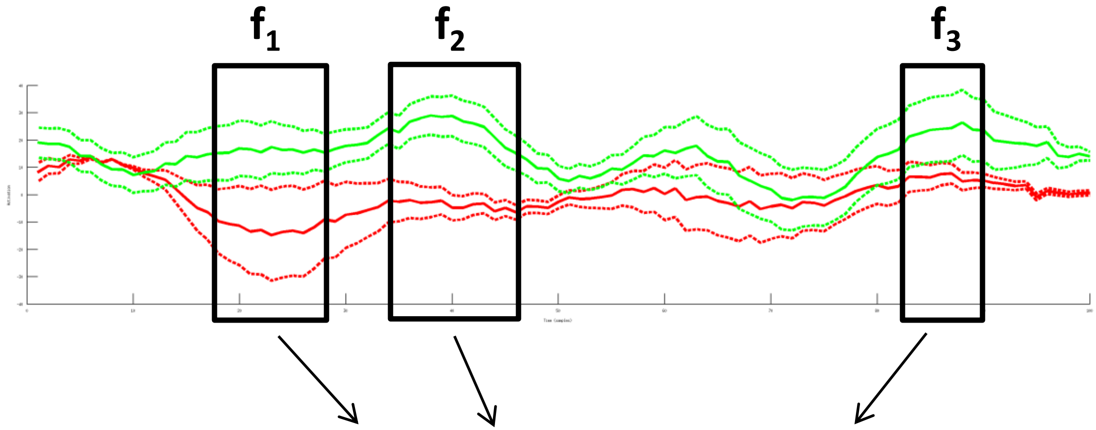


Resulting Segments

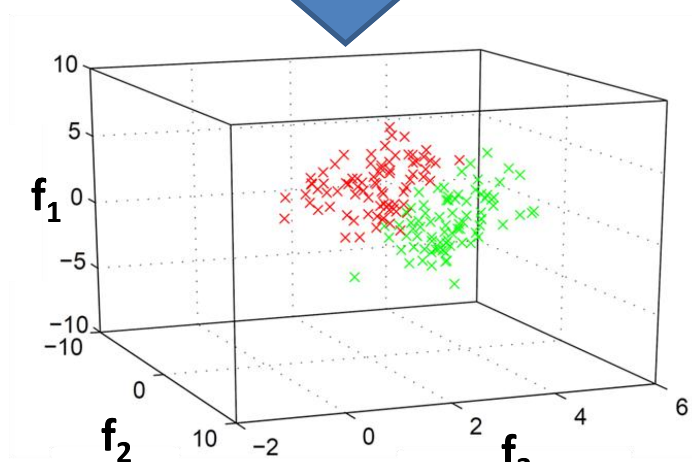


Extracting Key Features



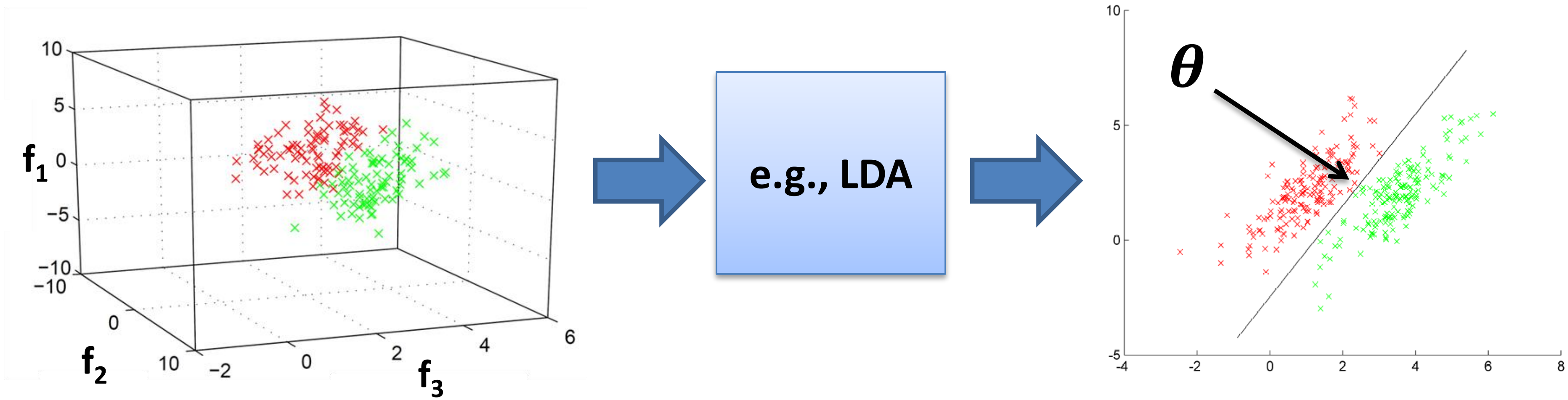


For each trial segment, calculate signal mean in 3 time sub-windows (→ 3-dim feature vector)



Using Machine Learning

- The feature vectors are passed on to a machine learning function (e.g., Linear Discriminant Analysis)



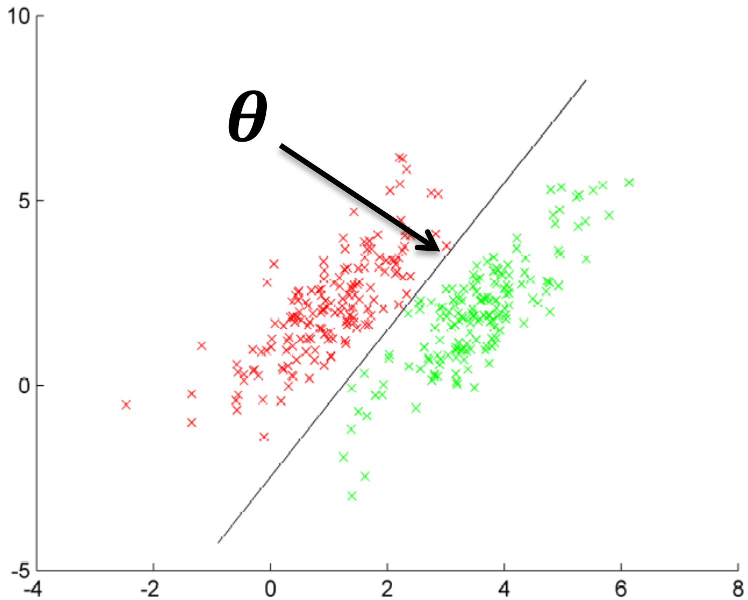
(Note: actually, this space has 3x #channels dimensions)

LDA In a Nutshell

- Given trial segments $\boldsymbol{x}_k$ (in vector form) in $\mathcal{C}_1$ and $\mathcal{C}_2$,

$$\boldsymbol{\mu}_i = \frac{1}{|\mathcal{C}_i|}\sum_{k\in\mathcal{C}_i} \boldsymbol{x}_k\,, \qquad \Sigma_i = \sum_{k\in\mathcal{C}_i} (\boldsymbol{x}_k - \boldsymbol{\mu}_i)(\boldsymbol{x}_k - \boldsymbol{\mu}_i)^\intercal$$

$$\boldsymbol{\theta} = (\Sigma_1 + \Sigma_2)^{-1}(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1), \qquad \mathrm{b} = \boldsymbol{\theta}^\intercal(\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2)/2$$



LDA In a Nutshell

- Given trial segments $\boldsymbol{x}_k$ (in vector form) in $\mathcal{C}_1$ and $\mathcal{C}_2$,

$$\boldsymbol{\mu}_i = \frac{1}{|\mathcal{C}_i|}\sum_{k\in\mathcal{C}_i}\boldsymbol{x}_k\,, \qquad \Sigma_i = \sum_{k\in\mathcal{C}_i}(\boldsymbol{x}_k-\boldsymbol{\mu}_i)(\boldsymbol{x}_k-\boldsymbol{\mu}_i)^\intercal$$

$$\boldsymbol{\theta} = (\Sigma_1+\Sigma_2)^{-1}(\boldsymbol{\mu}_2-\boldsymbol{\mu}_1), \qquad \mathrm{b} = \boldsymbol{\theta}^\intercal(\boldsymbol{\mu}_1+\boldsymbol{\mu}_2)/2$$

- **Caveat**: Σ_i often high-dimensional but only few trials available
- Can use a regularized estimator instead, here using *shrinkage*; instead of Σ_i, we use $\tilde{\Sigma}_i$ above:

$$\tilde{\Sigma}_i = (1-\lambda)\Sigma_i + \lambda\boldsymbol{I}$$

LDA In a Nutshell

- Given trial segments $\boldsymbol{x}_k$ (in vector form) in $\mathcal{C}_1$ and $\mathcal{C}_2$,

$$\boldsymbol{\mu}_i = \frac{1}{|\mathcal{C}_i|}\sum_{k\in\mathcal{C}_i}\boldsymbol{x}_k\,, \qquad \Sigma_i = \sum_{k\in\mathcal{C}_i}(\boldsymbol{x}_k - \boldsymbol{\mu}_i)(\boldsymbol{x}_k - \boldsymbol{\mu}_i)^\intercal$$

$$\boldsymbol{\theta} = (\Sigma_1 + \Sigma_2)^{-1}(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1), \qquad \mathrm{b} = \boldsymbol{\theta}^\intercal(\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2)/2$$

- Corresponding prediction function is linear in X:

$$\mathrm{y} = \mathrm{sign}(\boldsymbol{\theta}\,\mathrm{vec}(\boldsymbol{X})\; - b)$$

Linear Weights Visualized

- Color-coded linear weights topographies, 22 channels, 3 time windows, data from ERP task

How good is it?

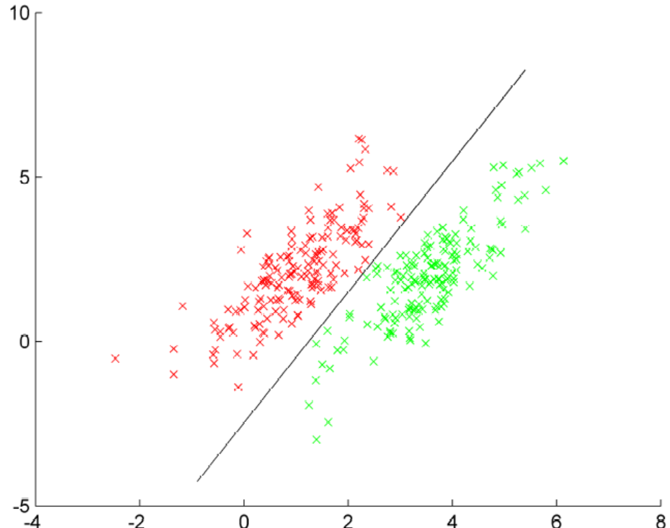
- Source activation S can be recovered from sensor measurements by a linear mapping if (linear) volume conduction is invertible ($\boldsymbol{S} = \boldsymbol{W}\boldsymbol{X}$)

How good is it?

- Source activation S can be recovered from sensor measurements by a linear mapping if (linear) volume conduction is invertible ($\boldsymbol{S} = \boldsymbol{W}\boldsymbol{X}$)
- Assuming a jointly Gaussian noise process and a noise distribution that is independent of the condition (A/B), LDA recovers the *optimal linear mapping*
- Shrinkage LDA on these features yields state-of-the-art ERP performance!

How good is it?

- Linear classifiers like LDA can operate implicitly on source ERPs, but:
  - EEG variation is often *not* Gaussian
  - Data variability *can* depend significantly on condition
  - For limited data samples, LDA is not necessarily optimal
  - Does not yield directly interpretable results



2. Complex Case

- Nonlinear operation in play, on *source* signals
- Due to, e.g., *shift indeterminacy* of source waveforms
  (no precise time-locking / jitter / high-frequency time course / …)
- **Oscillatory processes**: e.g., determining the amplitude of source oscillations

$S = W*X$ $F = \mathrm{abs}(\mathrm{DFT}(S))$ $y = \theta*F - b$

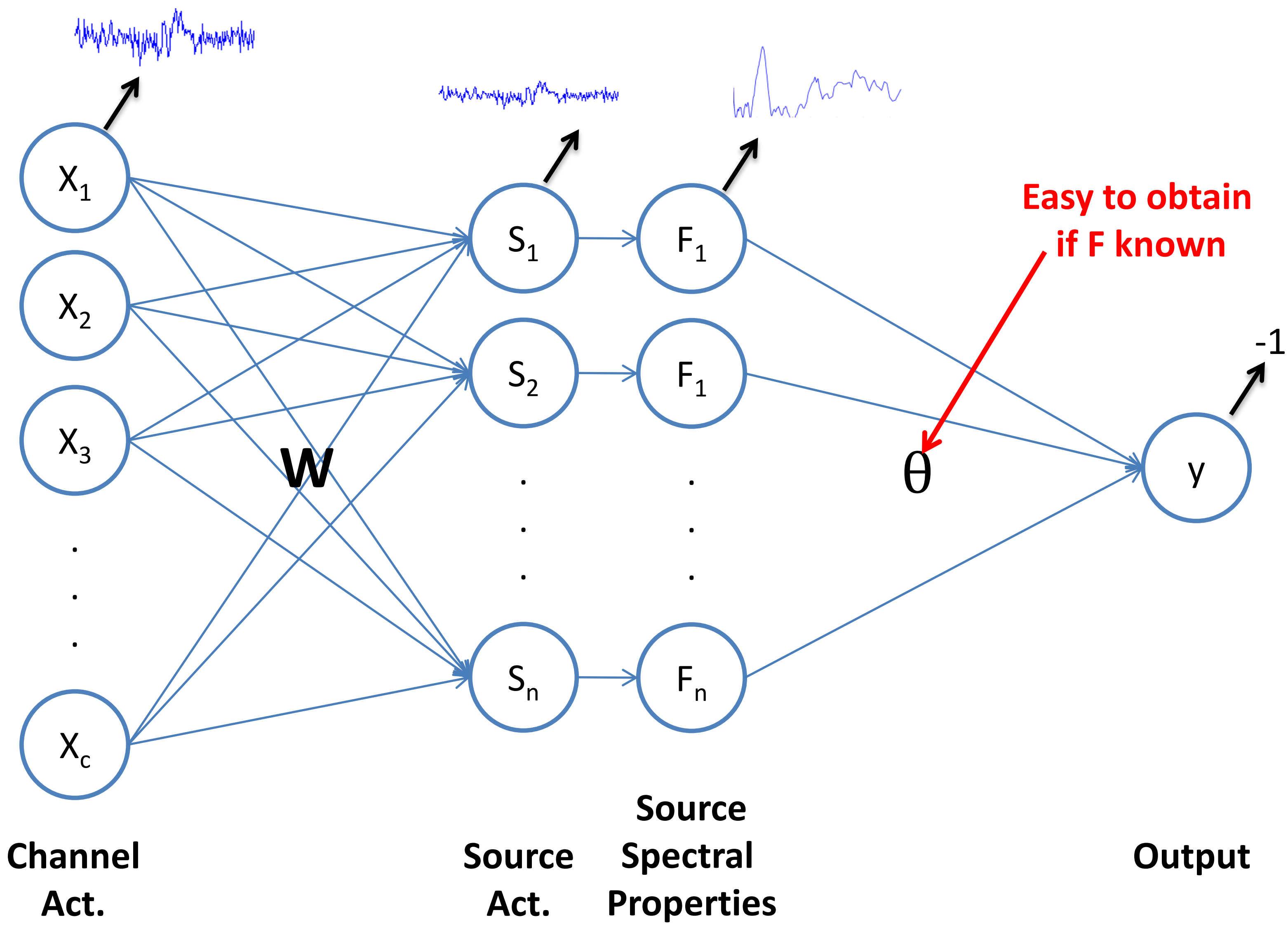
2. Complex Case

- Nonlinear operation in play, on *source* signals
- Due to, e.g., *shift indeterminacy* of source waveforms
  (no precise time-locking / jitter / high-frequency time course / …)
- **Oscillatory processes**: e.g., determining the amplitude of source oscillations

$S = W*X$ $\quad$ $F = \underbrace{\text{abs}}_{\text{nonlinear}}(\text{DFT}(S))$ $\quad$ $y = \theta*F - b$

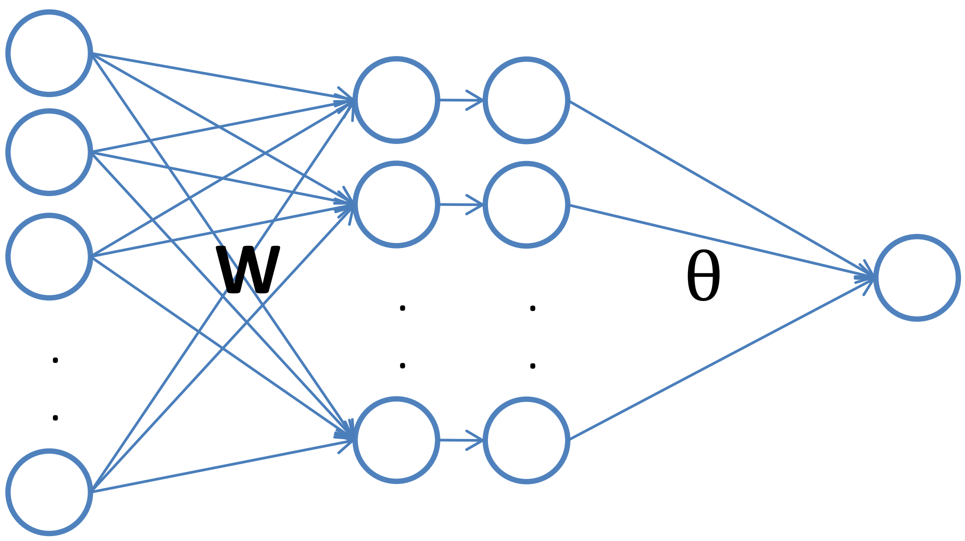
- Nonlinear and discards phase information
  (If done on channels, source spectral properties cannot be recovered)

Latent Variable Viewpoint



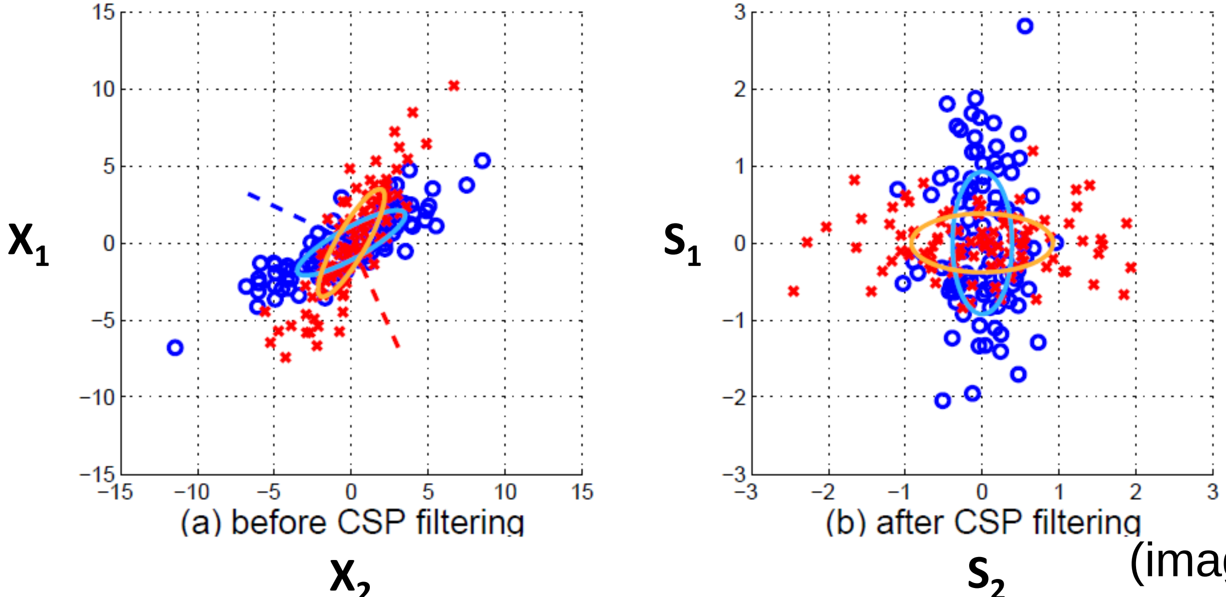
Latent Variable Viewpoint

- How to learn W?
  - "top-down" (using X & y) – gradient descent / NN backprop, …
  - "bottom-up" (using only X) – ICA, dictionary learning, …
  - both? – possibly supervised ICA, Bayesian inference, …
  - via direct observations (MR image, FW model) – Beamforming, …
  - using additional constraints (e.g., Gaussian signals) – CSP, DAL, …



Supervised Estimation

- Common Spatial Patterns
  - Most popular algorithm in BCI field for oscillatory processes
  - Assumption: **Gaussian**-distributed Signal, variance features (thus all structure captured by signal covariance)
  - Signal usually pre-filtered to known frequency band



(image: Blankertz 2009)

Supervised Estimation

- Common Spatial Patterns

  Given signal covariance matrix $\boldsymbol{\Sigma}_i$ under condition i,
  find the simultaneous diagonalizer **V** of $\boldsymbol{\Sigma}_1$ and $\boldsymbol{\Sigma}_2$

$$\boldsymbol{V}^{\top}\boldsymbol{\Sigma}_1\boldsymbol{V} = \boldsymbol{\Lambda}_1,$$
$$\boldsymbol{V}^{\top}\boldsymbol{\Sigma}_2\boldsymbol{V} = \boldsymbol{\Lambda}_2,$$

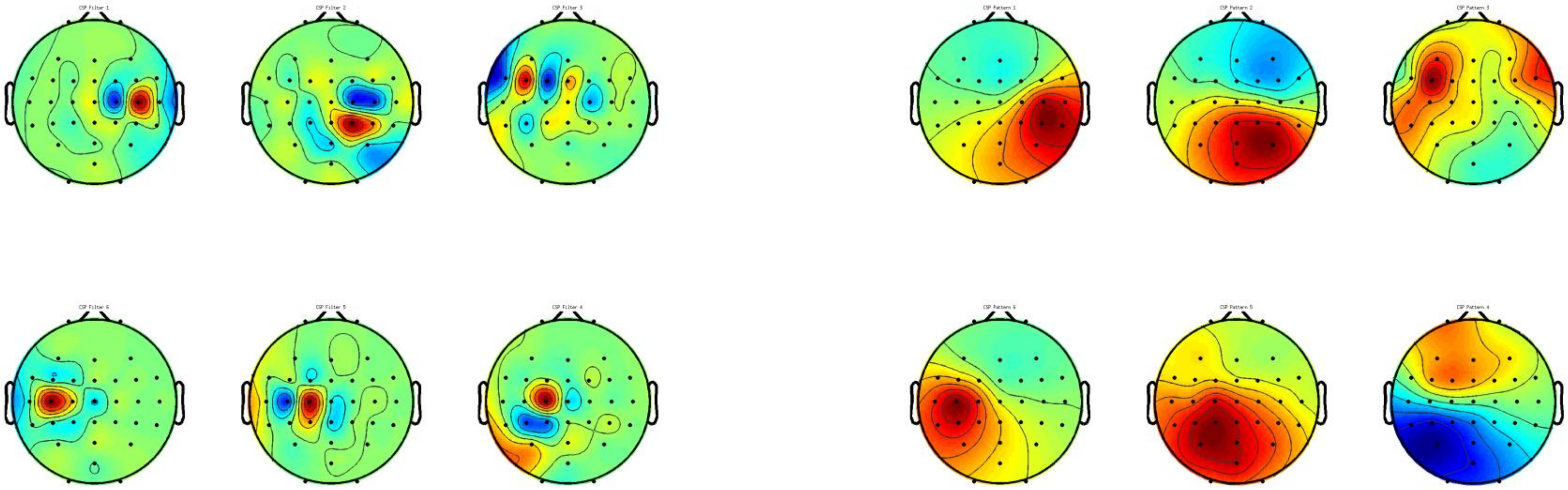
  (with $\boldsymbol{\Lambda}_i$ diagonal) such that $\boldsymbol{\Lambda_1} + \boldsymbol{\Lambda_2} = \boldsymbol{I}$. This yields a generalized eigenvalue problem of the form

$$\boldsymbol{V}^{\top}\boldsymbol{\Sigma}_1\boldsymbol{V} = \boldsymbol{D} \ \wedge \ \boldsymbol{V}^{\top}(\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2)\boldsymbol{V} = \boldsymbol{I}$$

  The k smallest and largest eigenvalues in **D** correspond to directions in **V** (spatial filters) that yield smallest (largest) variance in class 1 and simultaneously largest (smallest) variance in class 2.

Supervised Estimation

- Produces well-adapted filters (left) and occasionally roughly dipolar filter inverses (right)

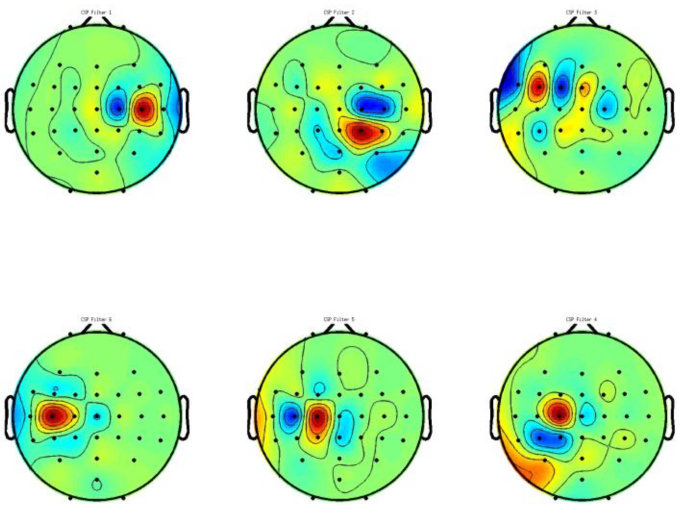


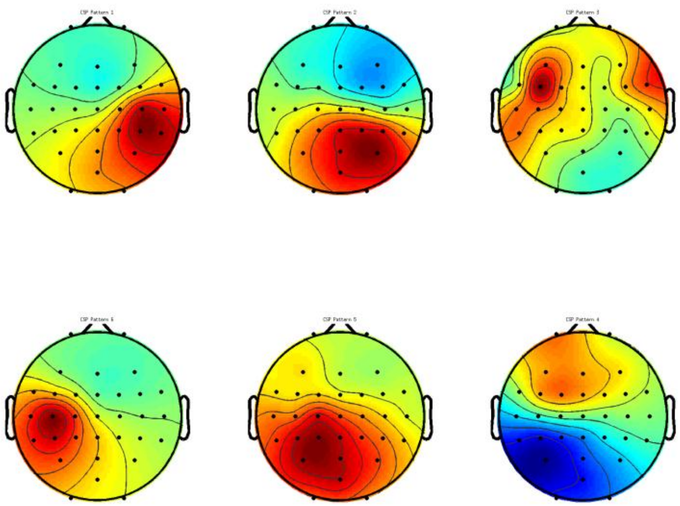
Complete CSP functional form:

$$y = \operatorname{sign}(\boldsymbol{\theta}\log(\operatorname{var}(\boldsymbol{W}\boldsymbol{X})) + b)$$

Supervised Estimation

- Produces well-adapted filters (left) and occasionally roughly dipolar filter inverses (right)





Complete CSP functional form:

$$y = \mathrm{sign}(\boldsymbol{\theta}\log(\mathrm{var}(\boldsymbol{W}\boldsymbol{X})) + b)$$

Usually learned via LDA

Outline

- Background
  - What is a BCI
  - What is BCILAB
- Theory
  - Overview
  - ERP approaches
  - Oscillatory approaches
- Practice
  - Toolbox overview
  - GUI & scripts walkthrough

BCILAB Components

Interface

GUI / Scripting Interfaces

- **Approach Definition**
- **Online Execution**
- **Offline Evaluation**
- **Visualization**

Methods

Signal Processing

ICA, SSA, FIR, IIR, FFT, …

Machine Learning

LDA, QDA, DAL, GMM, SVM, …

BCI Paradigms

CSP, Spec-CSP, ERP, RSSD, …

Devices

TCP, OSC, BCI2000, …

Infrastructure

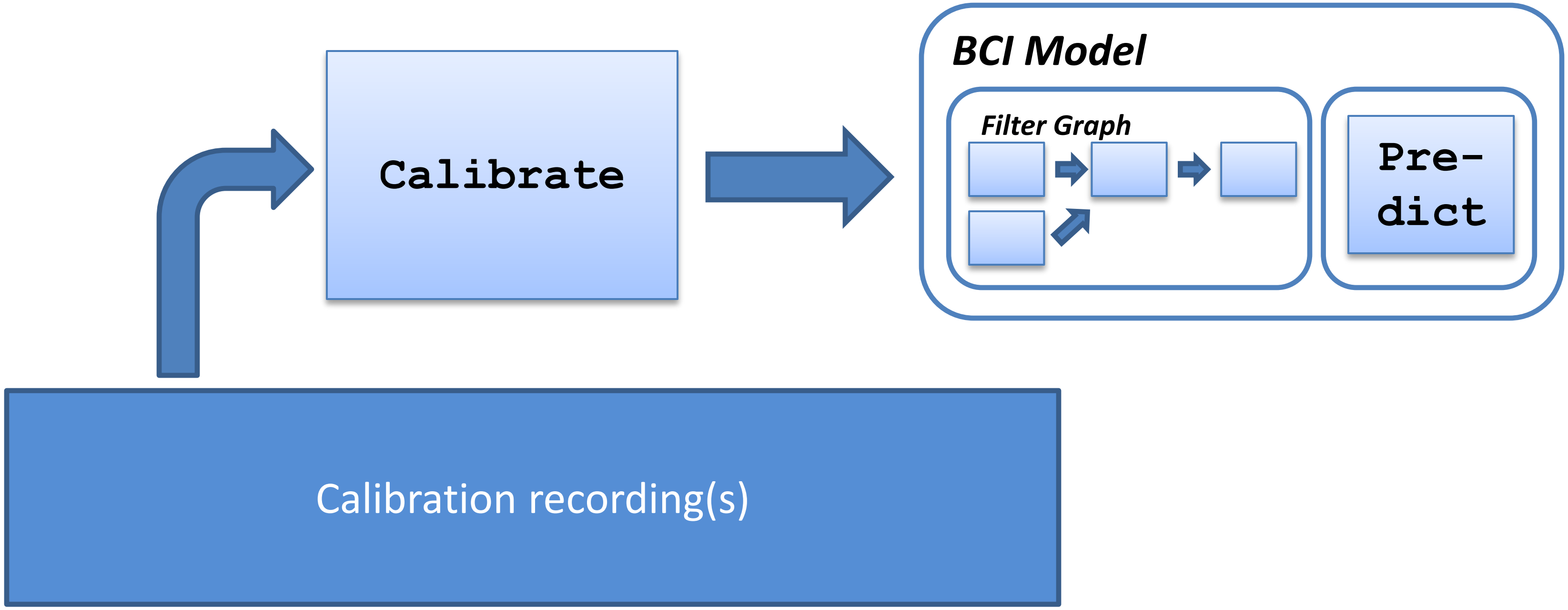
GUI generation, cluster computing, disk caching, helper functions, environment services

Dependencies

CVX, BNT, EEGLAB, GUI utils, LIBSVM, GLMNET, …, Driver I/O

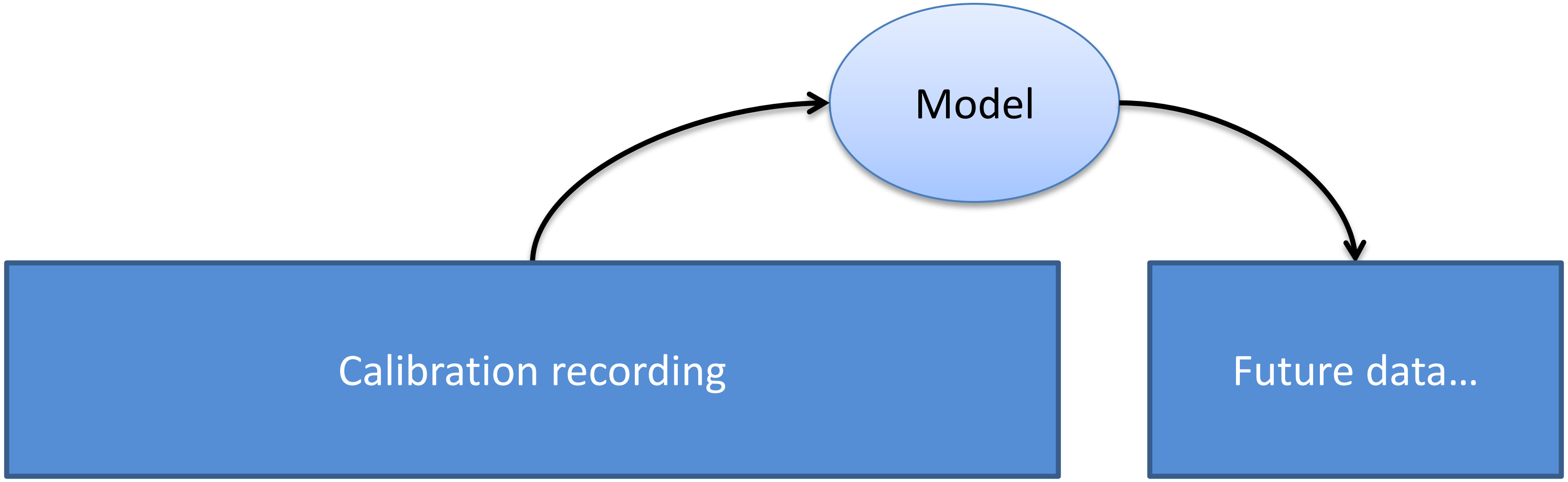
BCI Paradigms?

- BCI paradigms are the coarsest plugin type in BCILAB and tie all parts of a BCI approach together (signal processing, feature extraction, machine learning, …)
- They often generalize well to new BCI designs



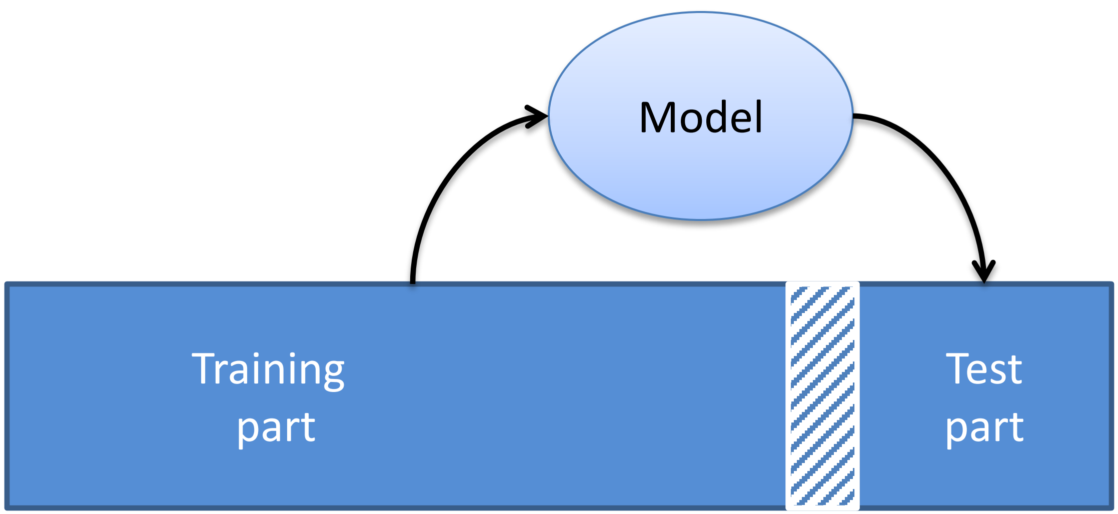
Evaluating Models

- Given calibration data
  - Estimate model parameters (spatial filters, statistics)
  - Apply the model to new data (online / single-trial)
- Optionally: compare outputs with known state, compute loss statistics for the model / approach (e.g., mis-classification rate)



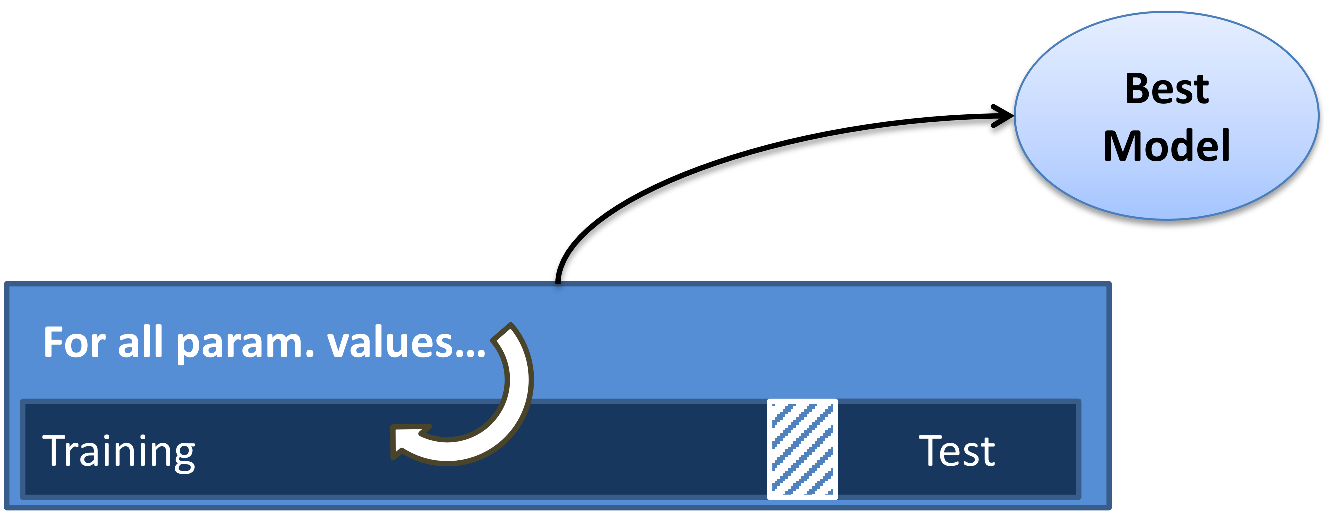
Evaluating Models

- Evaluation of computational approaches on a **single** data set?
  - Can not test on the training data (always on separate data)
  - Instead can split data set repeatedly into training/test blocks systematically, a.k.a. *cross-validation*



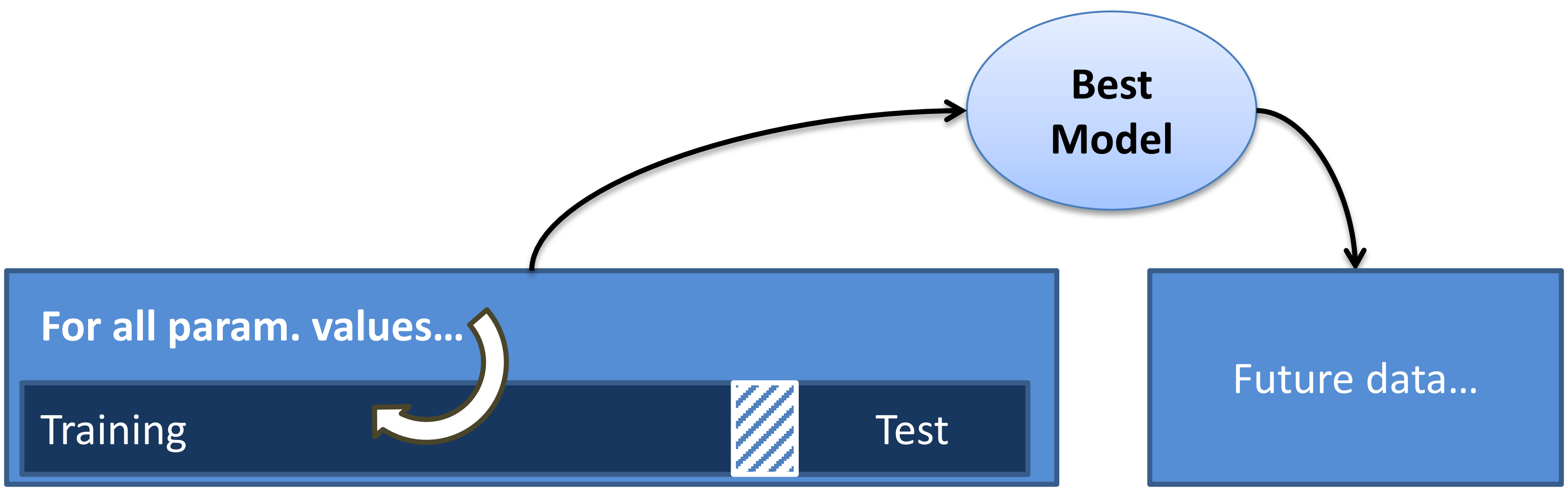
Resolving Free Parameters

- Can be done using cross-validation in a grid search (try all values of free parameters)
- **Caveat**: Resulting “optimal” numbers are *non-reportable* (cherry-picked!)



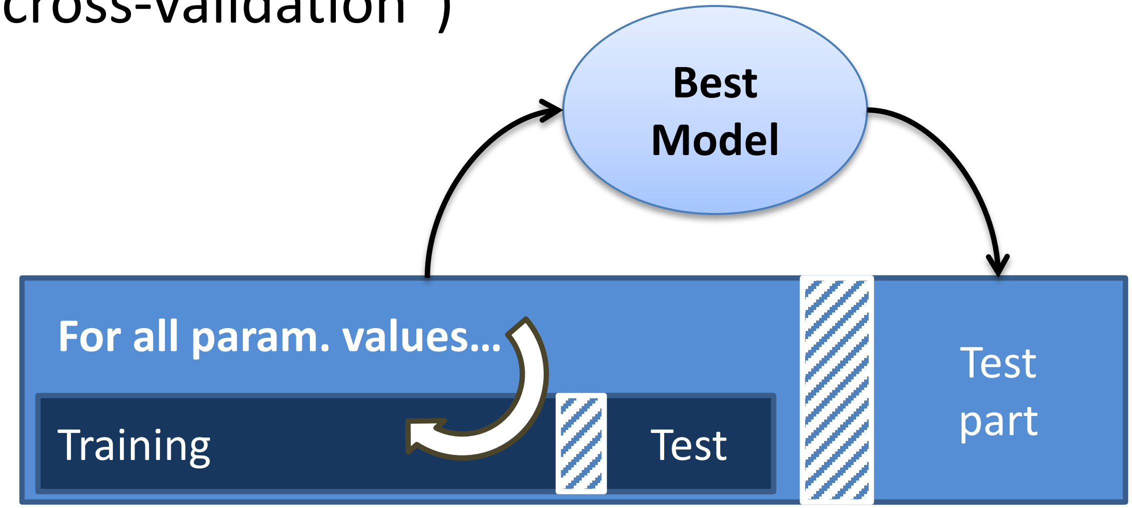
Resolving Free Parameters

- Can be done using cross-validation in a grid search (try all values of free parameters)
- **Caveat**: Resulting "optimal" numbers are *non-reportable* (cherry-picked!)
- But may test resulting best model on separate data

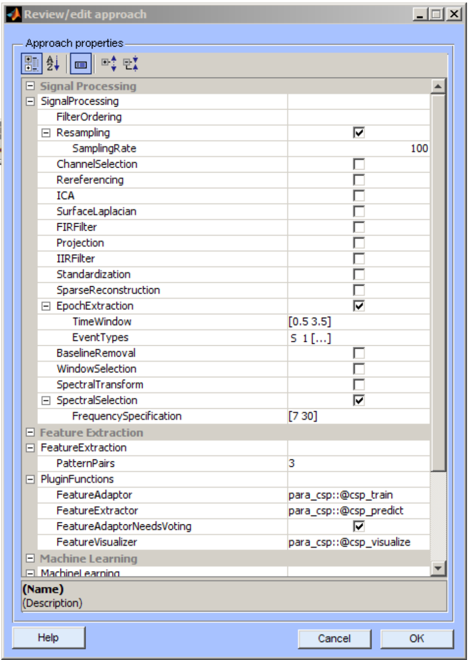


Resolving Free Parameters

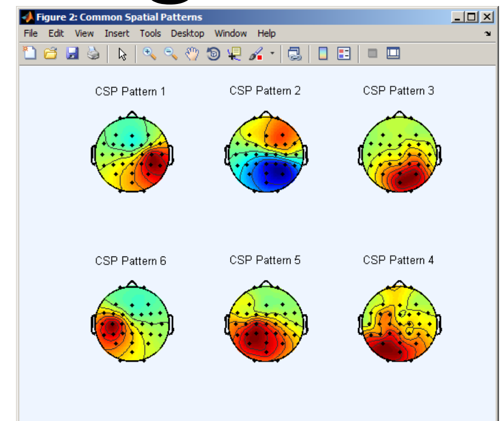
- Can be done using cross-validation in a grid search (try all values of free parameters)
- **Caveat**: Resulting "optimal" numbers are *non-reportable* (cherry-picked!)
- But may test resulting best model on separate data
- **Or** run grid search *within* an outer cross-validation ("nested cross-validation")

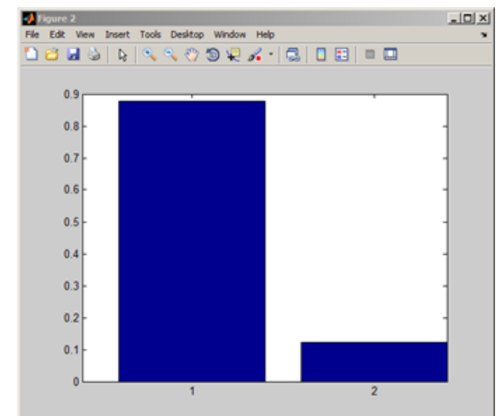


GUIs & Scripting Walkthrough



Hands-On Demo

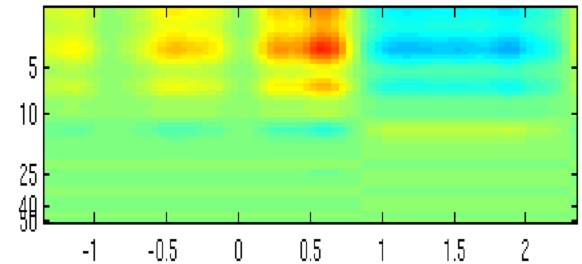




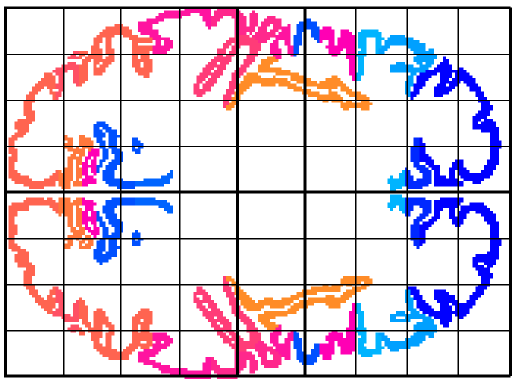

```
% define markers; here, two groups of markers are being defined; the first group represents class 1
% (correct responses), and the second group represents class 2 (incorrect responses).
mrks = {{'S101','S102'},{'S201','S202'}};
wnds = [0.25 0.3;0.3 0.35;0.35 0.4; 0.4 0.45;0.45 0.5;0.5 0.55;0.55 0.6];

% define approaches
approaches.wmeans_lda = {'Windowmeans' 'flt',{'events',mrks,'epoch',[0 0.8],'spectrum',[0.1 15]},'fex',{'wnds',wnds}};
approaches.wavelet_lars = {'Dataflow' 'flt',{'events',mrks,'epoch',[0 0.8],'spectrum',[0.1 15],'wavelet','on'},...
    'ml',{'learner',{'logreg',[],'variant','lars'}}};
approaches.dal = {'DAL_Lofreq','SignalProcessing',{'Resampling',60,'IIRFilter','off','FIRFilter',[0.1 0.5 18 21], ...
    'EpochExtraction',{'EventTypes',mrks,'TimeWindow',[-0.2 0.65]}},'MachineLearning',{'Learner',{'dal',2.^(8:-0.125:1)}}};

% run a batch analysis...
results = bci_batchtrain('Datasets','/data/projects/grainne/ERN/*.vhdr','Approaches',approaches,'RetainExistingResults',true);
```

Current Research

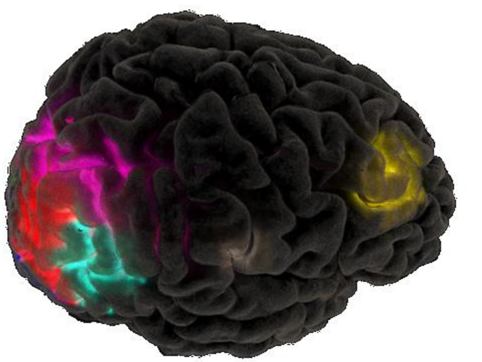
- More structural prior knowledge
  - E.g., smoothness/coupling, structured sparsity, kernels, dictionaries, per-trial parameters (e.g. ,“outlyingness”, shift)



- Quantitative prior knowledge
  - Structure atlases (Talairach, LONI, …) can supply information about the *a priori* relevance of a brain process



- Empirical prior knowledge
  - Data collected from other subjects can be co-registered/aligned and yield empirical prior distributions



Thanks!

Questions?