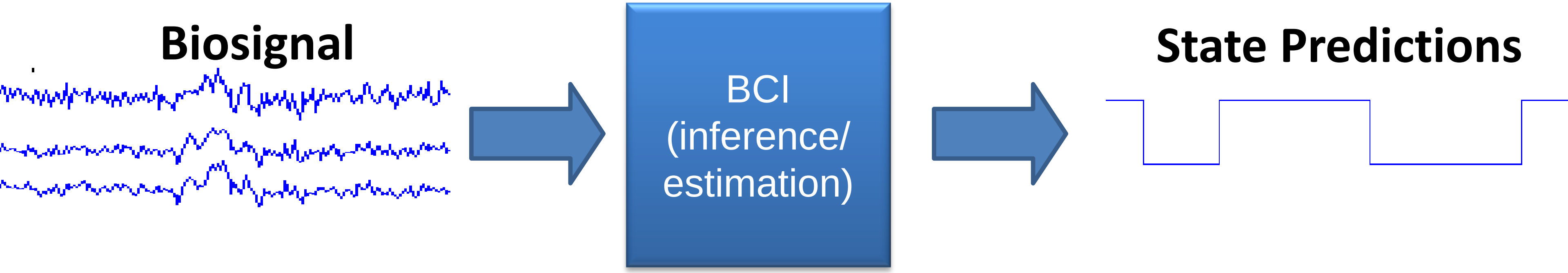
Statistical Learning Theory and Brain-Machine Interface Design

Christian A. Kothe

Swartz Center for Computational Neurscience, UCSD

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."
  - Biosignal: EEG, ECoG, MEG, ... (+ possibly non-brain data)
  - Abstract aspect of cognitive state: "*type of limb movement imagined*", "*degree of surprisal*", "*type of vowel imagined*"
  - (doesn't have to be properly defined for the BCI to work)



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

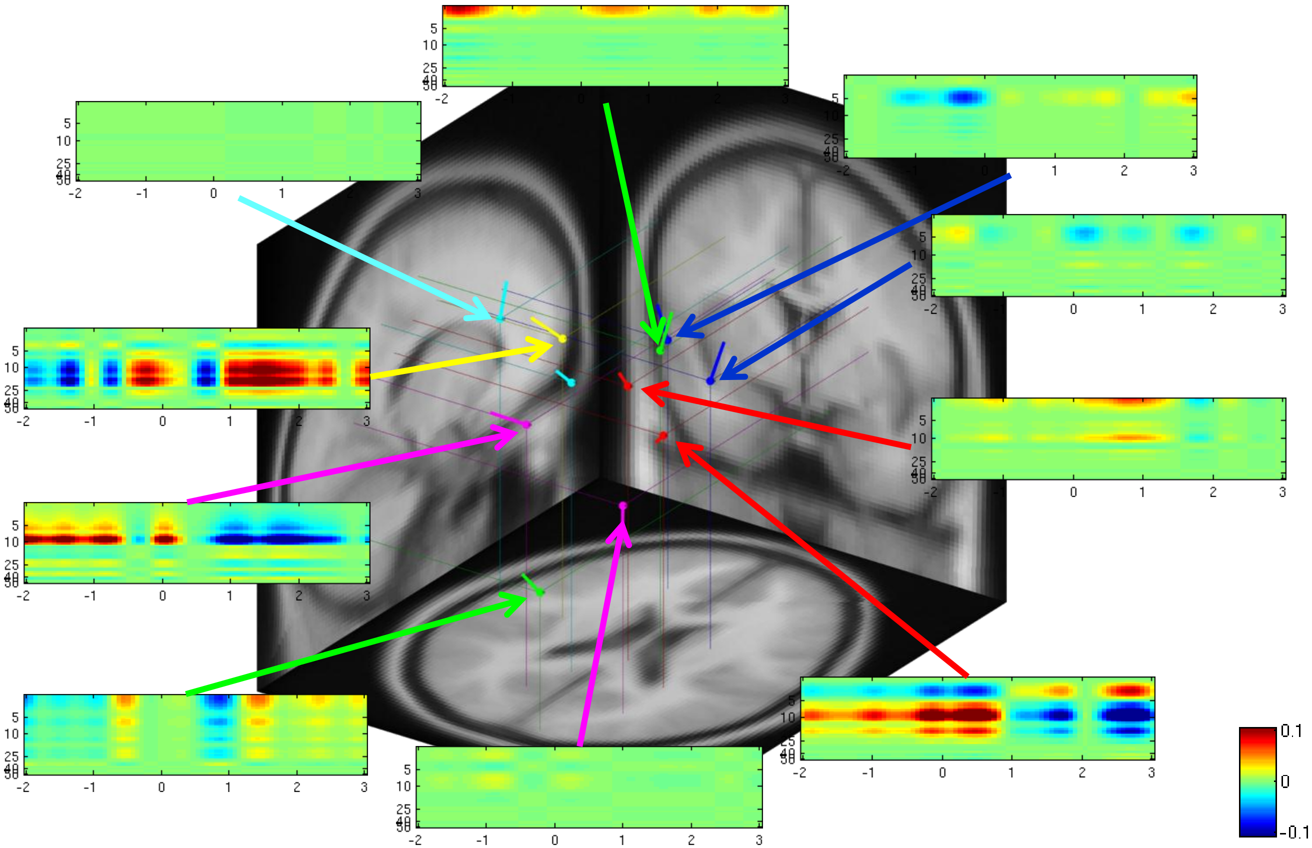
Research Directions

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



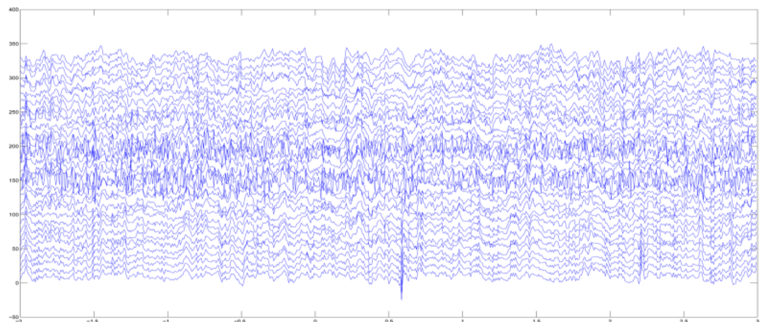
Research Directions

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



How does a BCI work?

- Mathematical mapping

y = f(**X**);   **X**= 

y= “left hand” (-1)
“right hand” (+1)

- Functional form

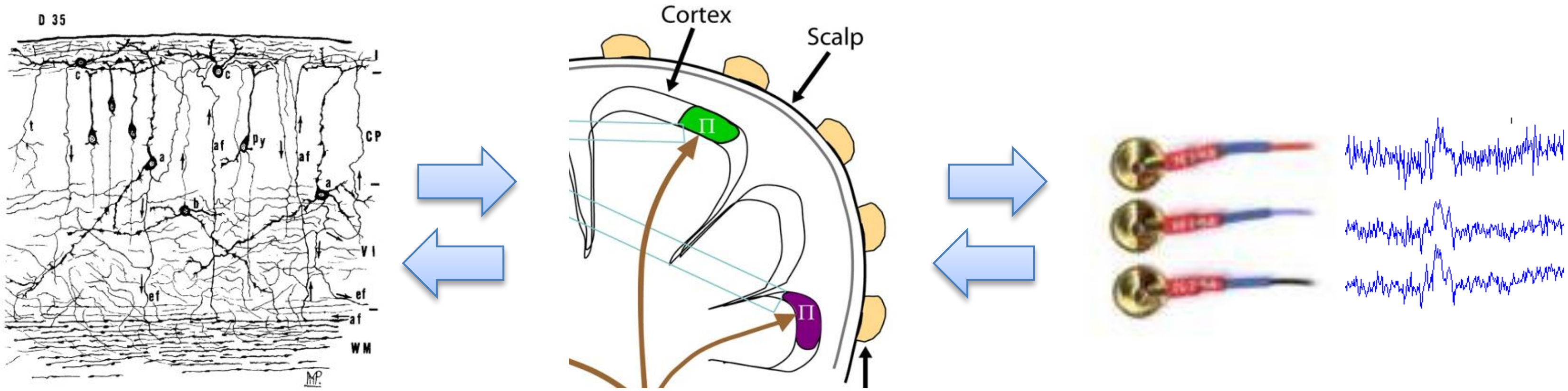
e.g., $y = \mathrm{sign}(\mathrm{var}(\boldsymbol{WX}) + b)$

- Unknown parameters!

e.g., **W**, b, …

Functional Form?

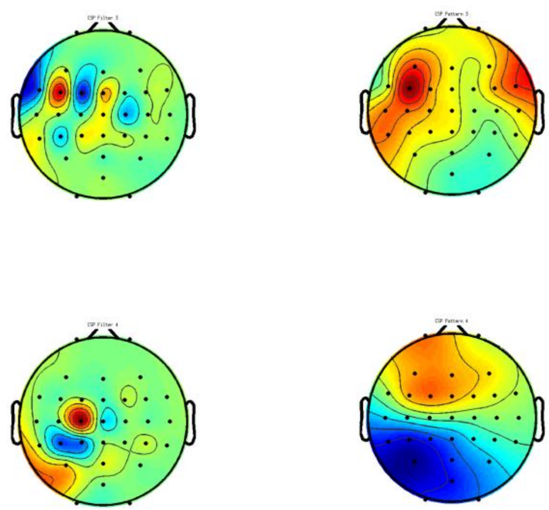
- Reflects the relationship between observation (data segment X) and desired output (cognitive state parameter y)
- Based on some assumed generative mechanism (forward model) or ad hoc



- Note: Functional form is the inverse mapping!

Basic Ingredient: Spatial Filter

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



W **A**

Further Ingredients

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

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

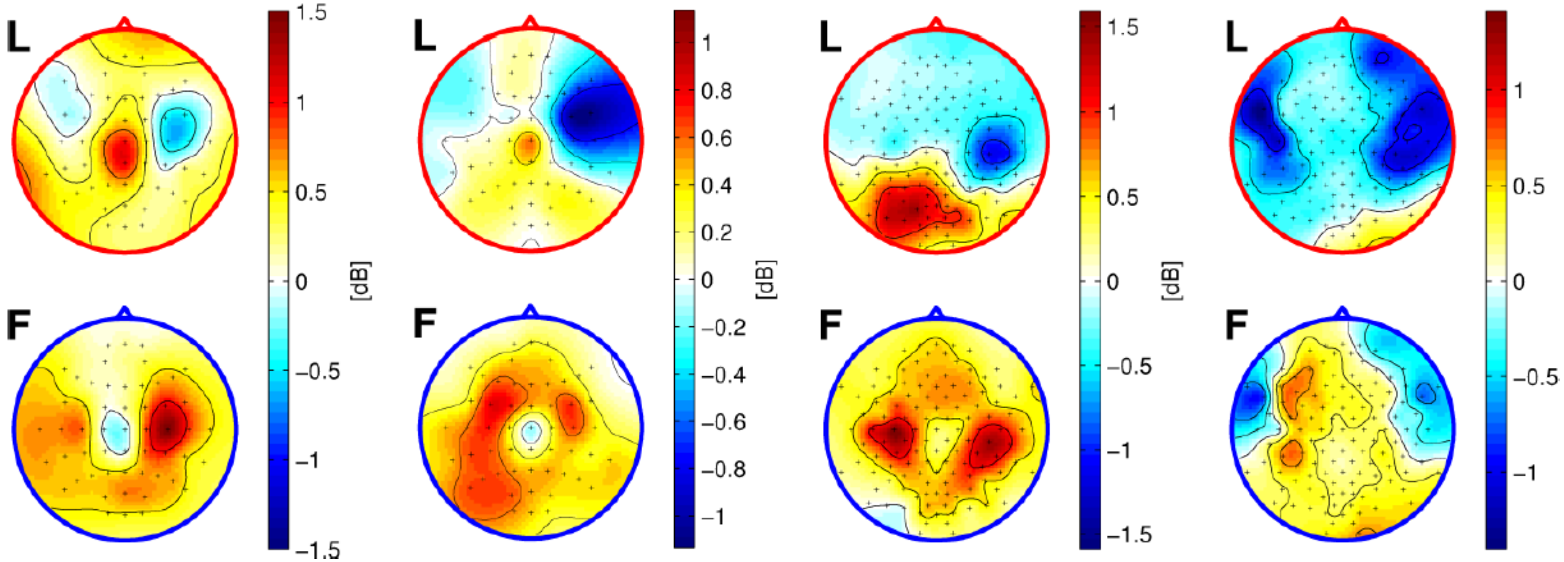
$$y = \boldsymbol{\theta}\,\mathbf{vec}(|(\boldsymbol{WX})\boldsymbol{T}|) + b$$ **(nonlinear…)**

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:

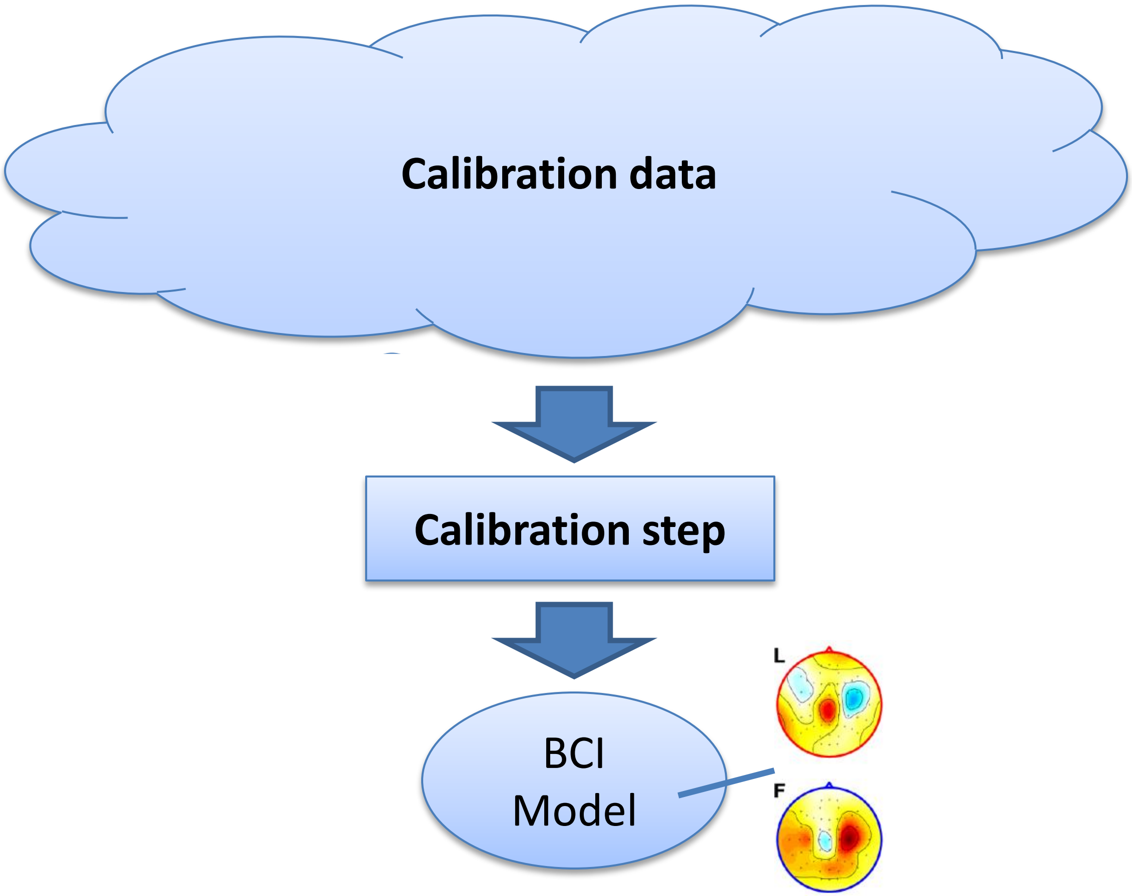


Person 1 Person 2 Person 3 Person 4

(image: Blankertz et al. 2007)

Model Calibration

- Need *calibration / training data* to estimate parameters from, and a separate *calibration step*

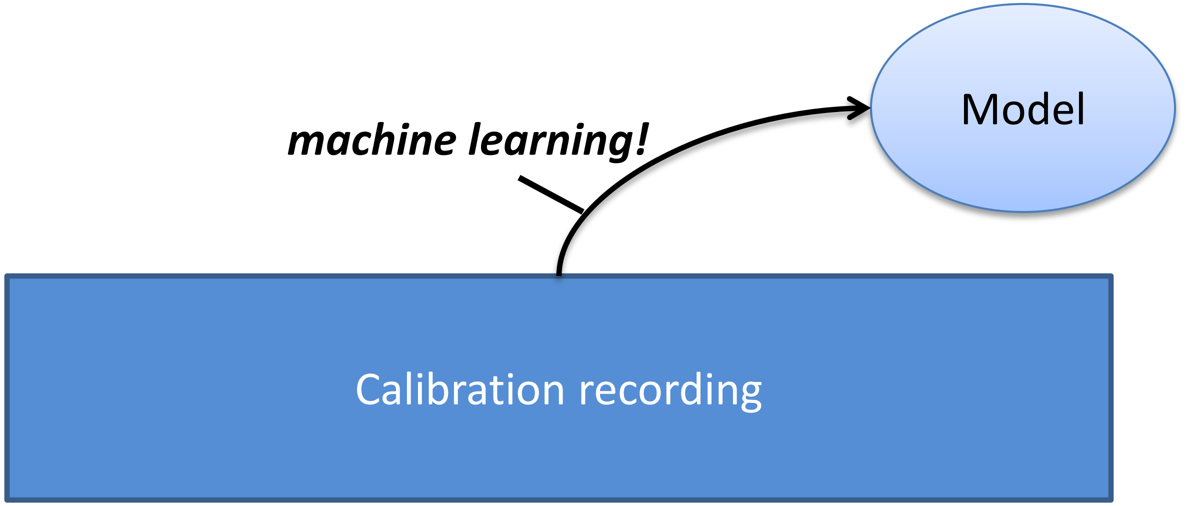


Model Calibration

- In theory many possibilities (e.g. MR scanner data + Beamforming)

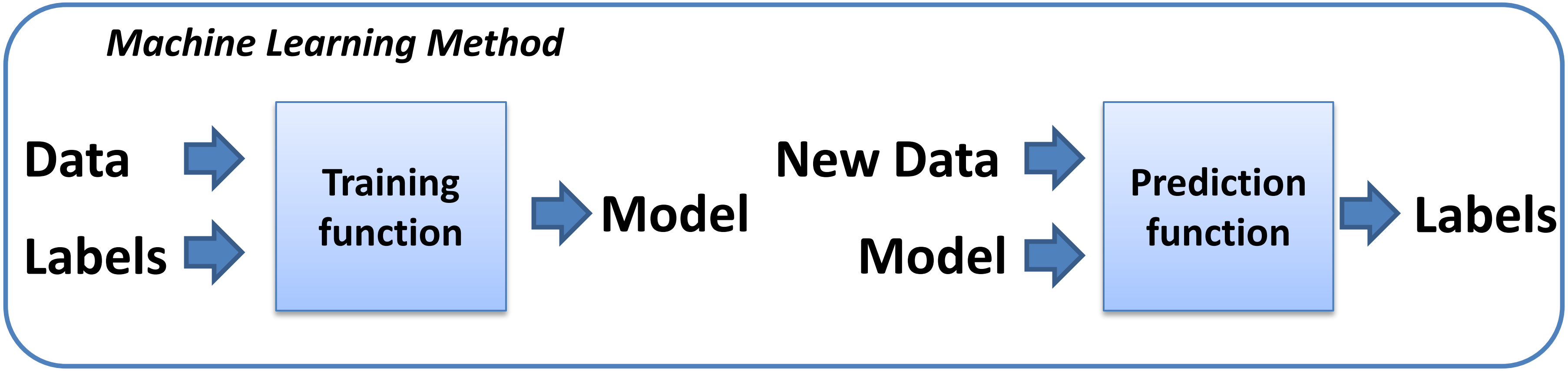
Model Calibration

- In theory many possibilities (e.g. MR scanner data + Beamforming)
- 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

- 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.

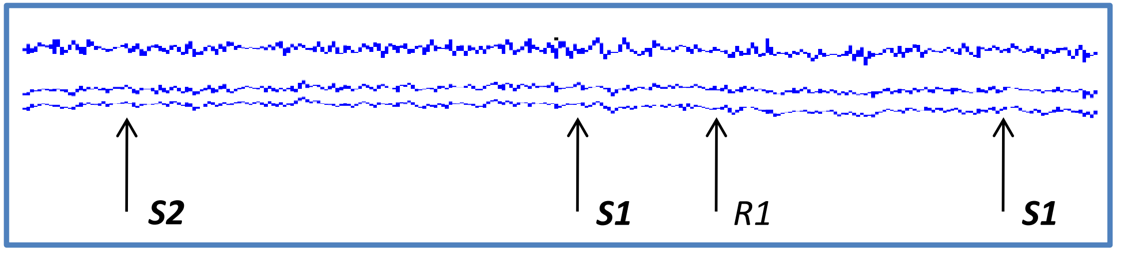


Required Calibration Recording

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

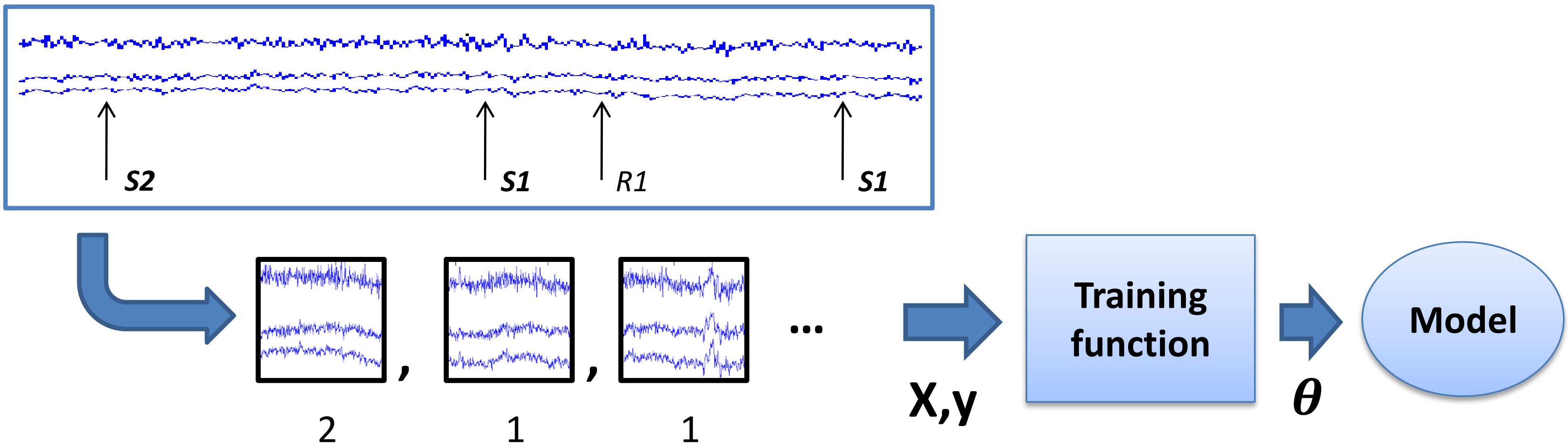
Required Calibration Recording

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



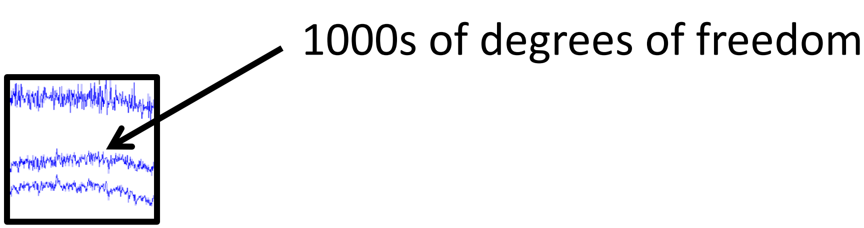
Using Machine Learning

- 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)
  - (but note: different story for custom methods)

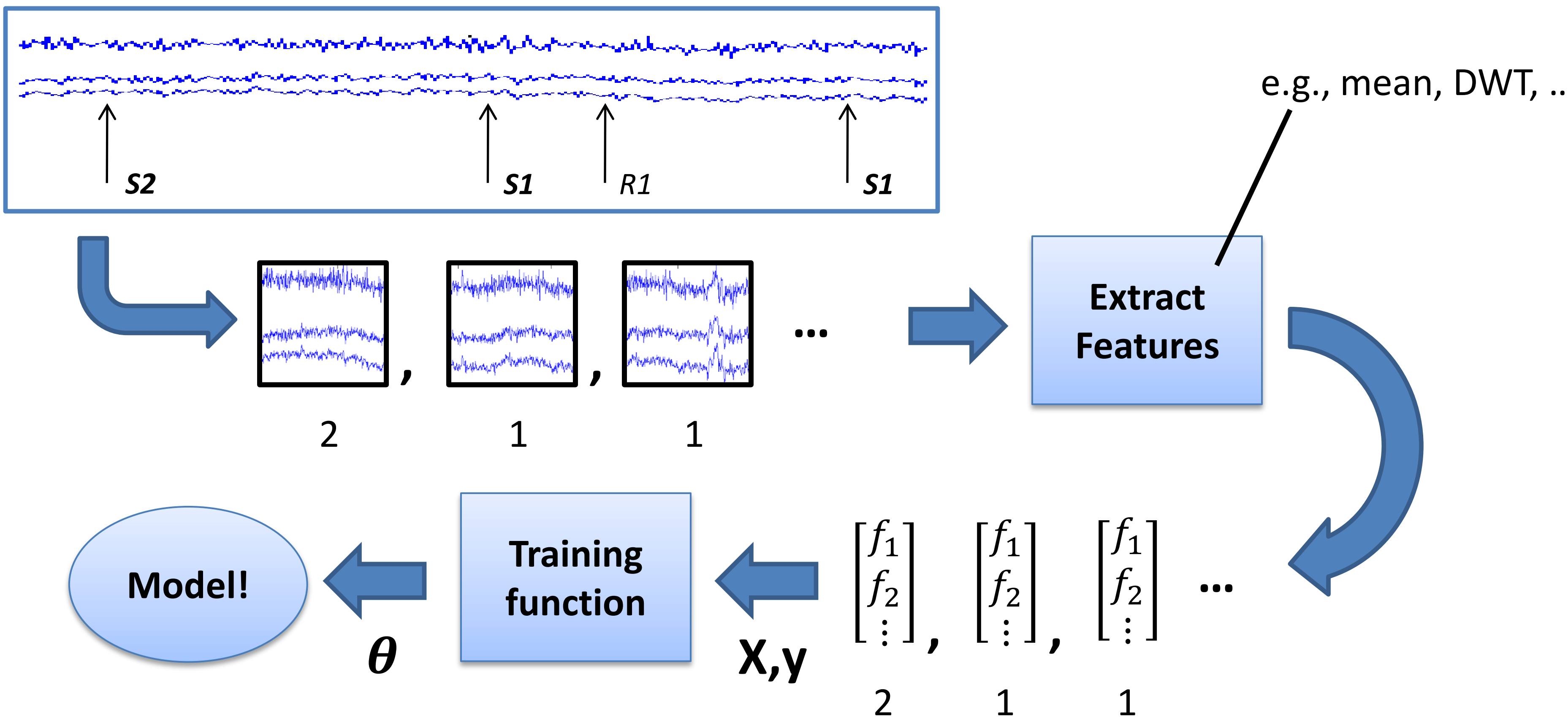


Detour: Feature Extraction

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

Using Machine Learning

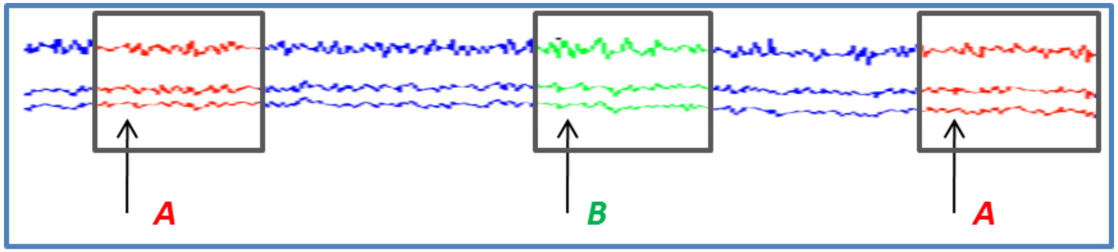
- Including feature extraction, the analysis process is as follows:



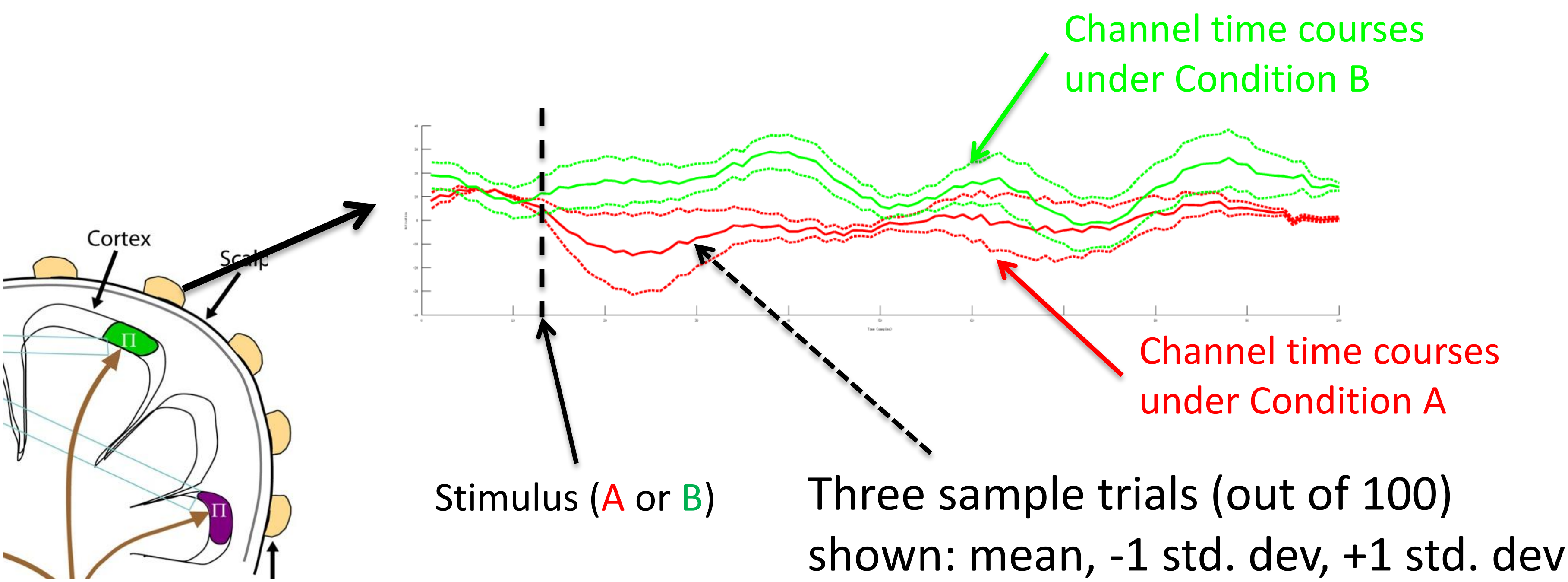
Two Major Analysis Pathways

Simple Case: ERP-like Patterns

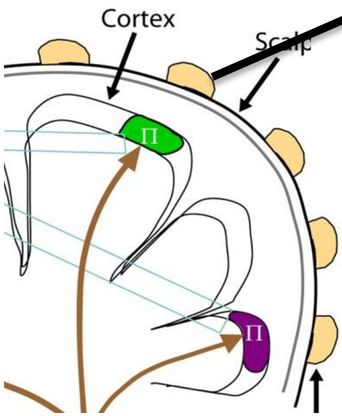
- Suppose a calibration recording with 100 stimuli of type A and 100 stimuli of type B

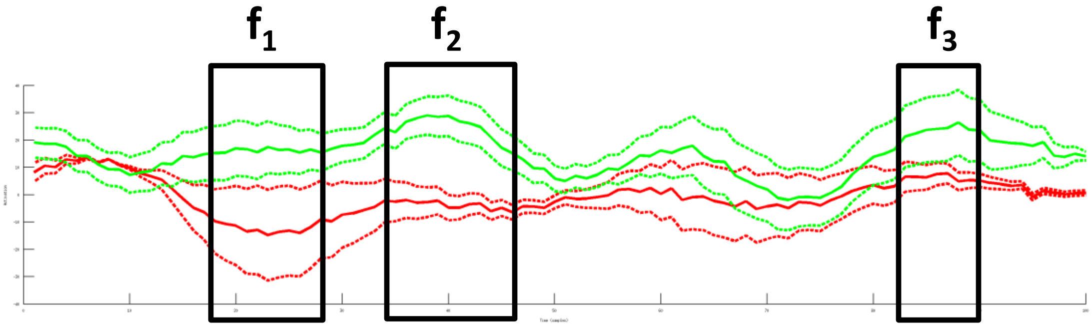


Resulting Segments

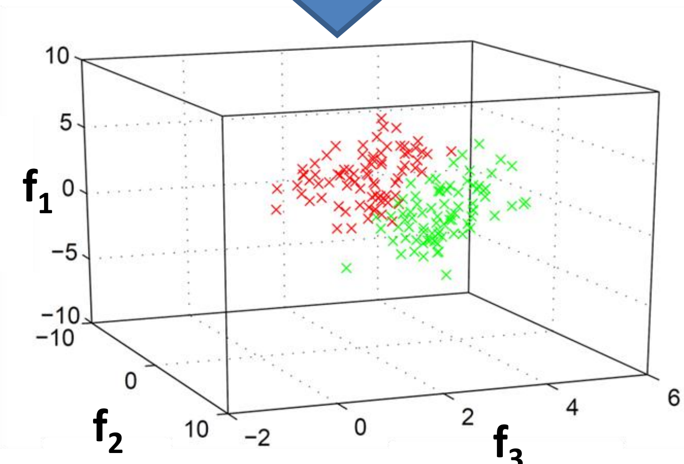


Extracting Key Features



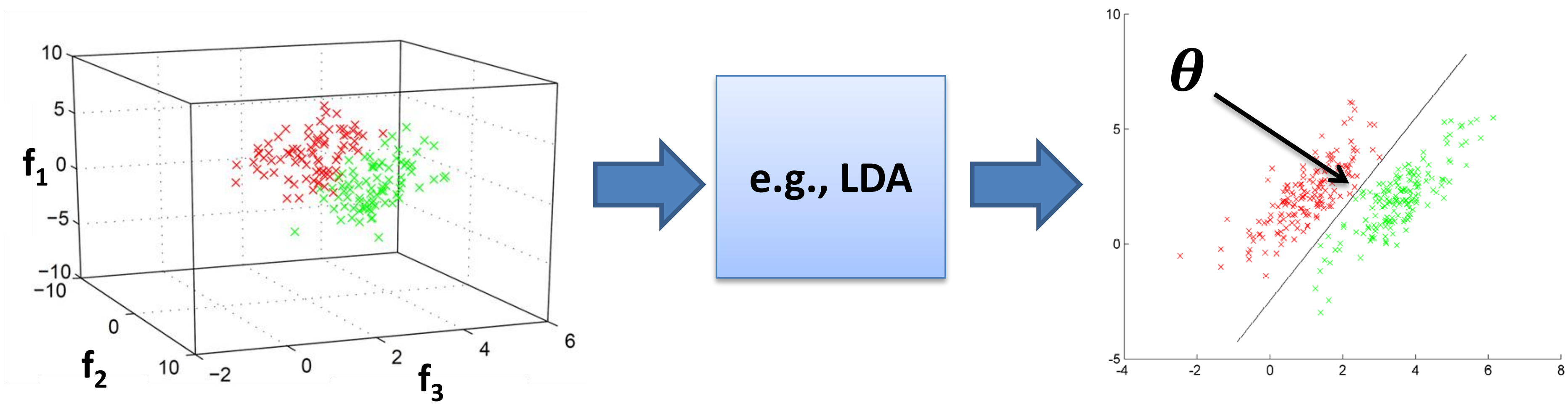


For each trial segment, calculate signal mean in 3 time sub-windows (→ 3-dim feature vector)



Using Machine Learning

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



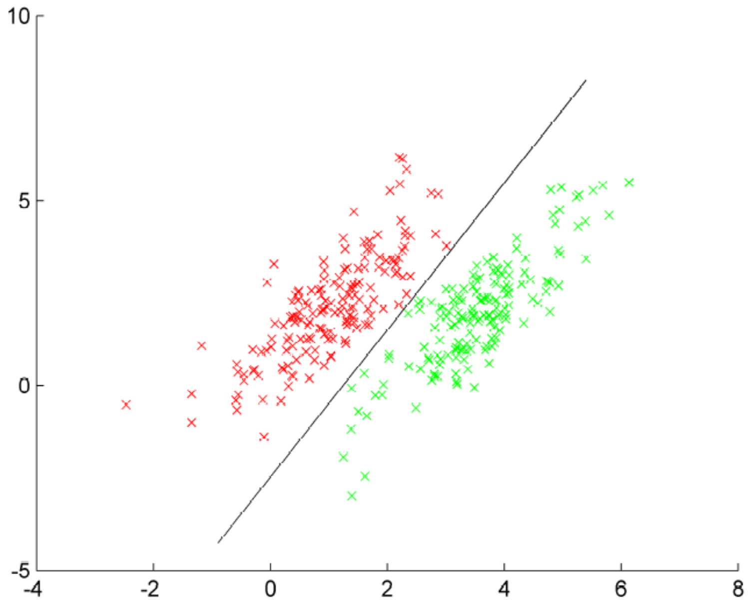
(Note: actually, this space has 3x #channels dimensions)

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**: θ 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 $\mathcal{C}_2$,

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

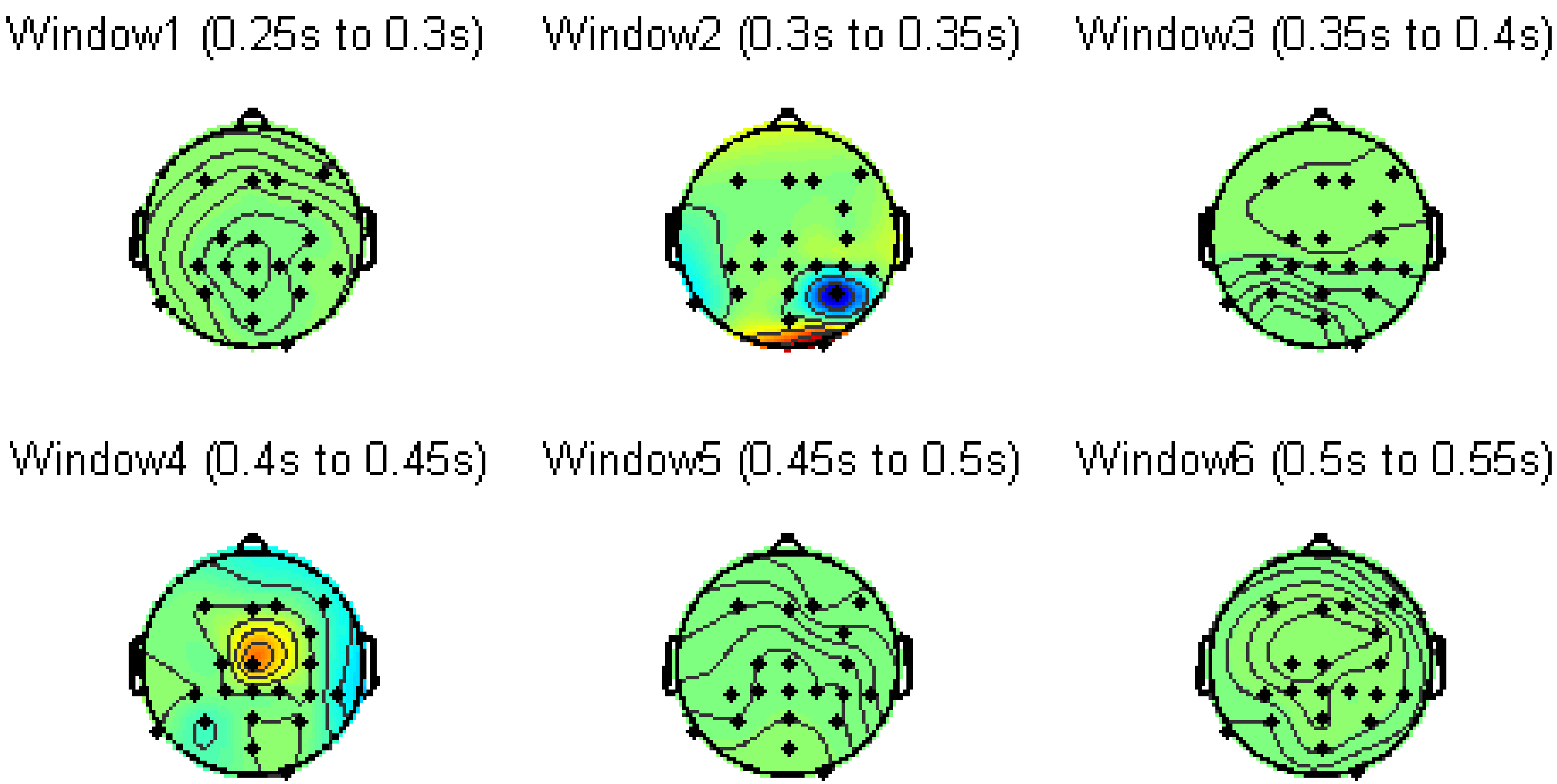
$$\boldsymbol{\theta} = (\Sigma_1 + \Sigma_2)^{-1}(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1), \qquad \mathrm{b} = \boldsymbol{\theta}^\intercal(\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2)/2$$

- Corresponding prediction function is linear in X:

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

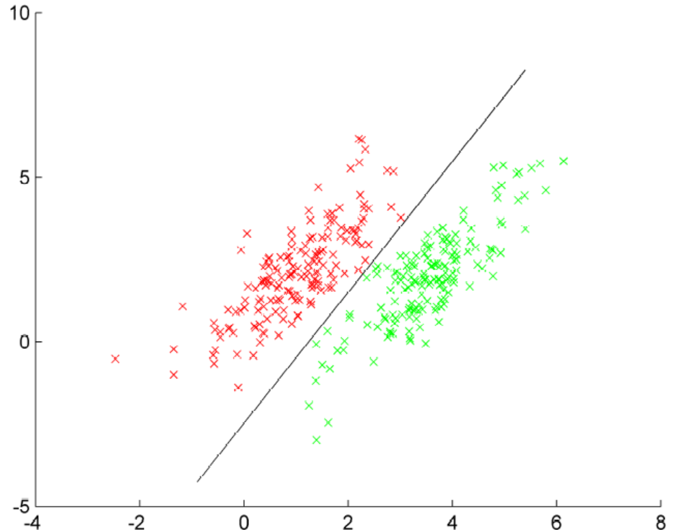
Linear Weights Visualized

- Color-coded linear weights topographies, 22 channels, 6 time windows, data from ERP task



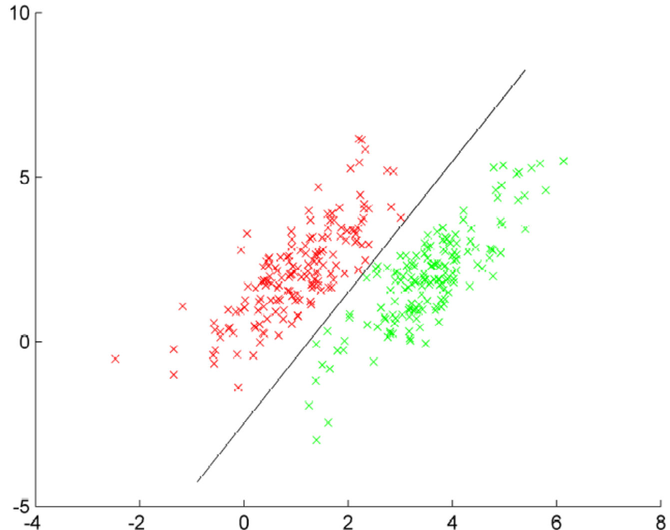
Does it Make Sense?

- 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*



Does it Make Sense?

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

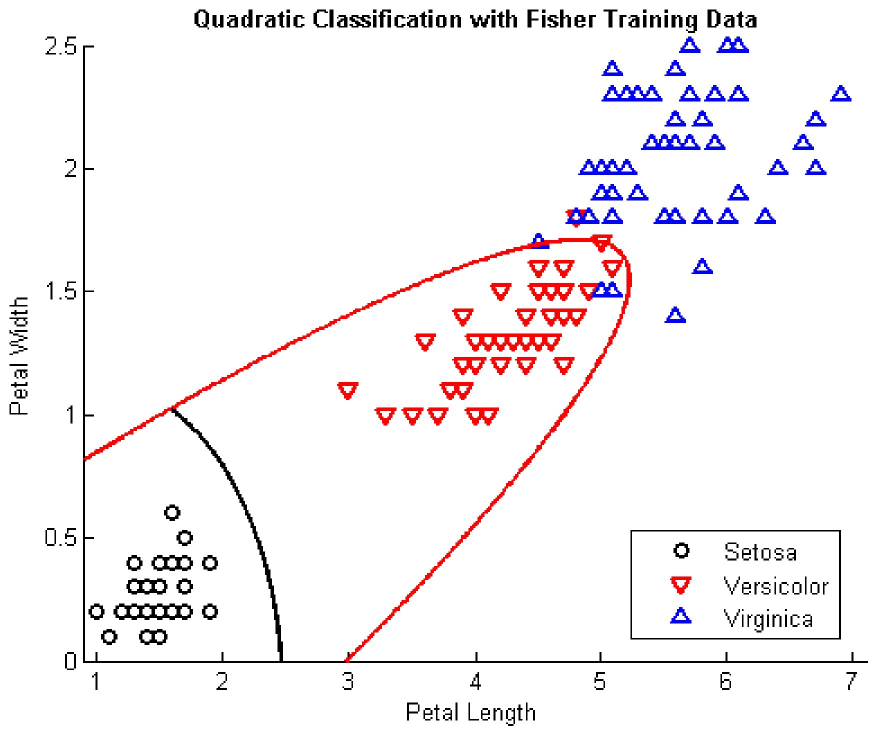


Does it Make Sense?

- Linear classifiers like LDA can operate implicitly on source ERPs, but:
  - EEG variation is often *not* Gaussian
  - Data variation *can* depend significantly on condition
  - For limited data samples, LDA is not necessarily optimal
  - Does not yield directly interpretable results
- Also in the linear framework:
  - Using the full source activation segments instead of their mean features
  - Using source wavelet features

Digression: Alternatives

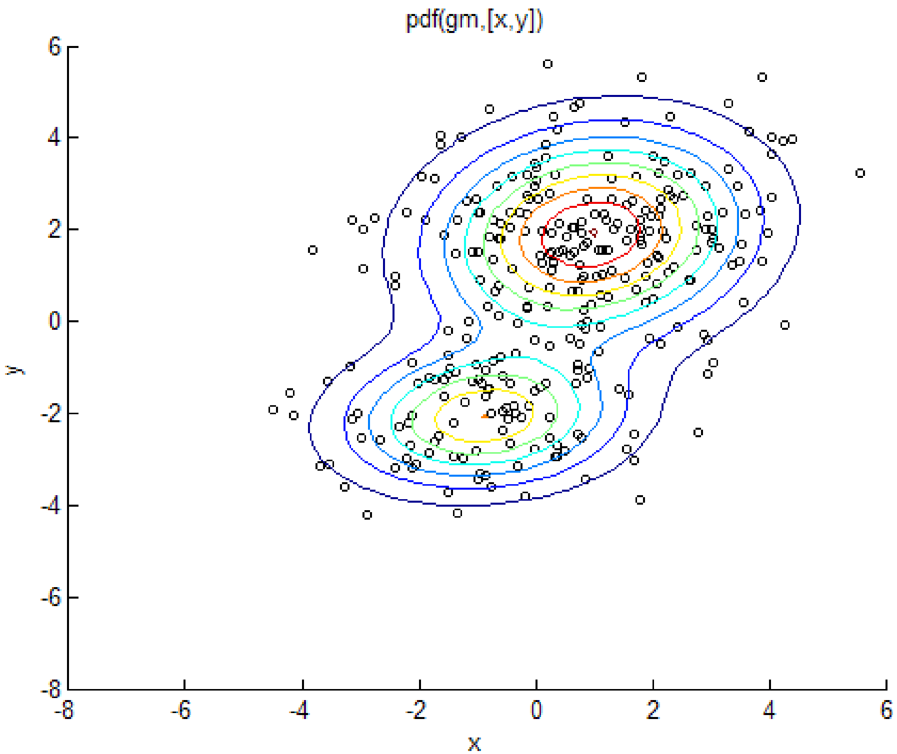
- Omitting the assumption of condition-independent noise yields Quadratic Discriminant Analysis (QDA)



(image: The Mathworks)

Digression: Alternatives

- Fitting multiple Gaussians for each condition instead of one yields Gaussian Mixture Models



(image: The Mathworks)

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

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

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

Complex Case

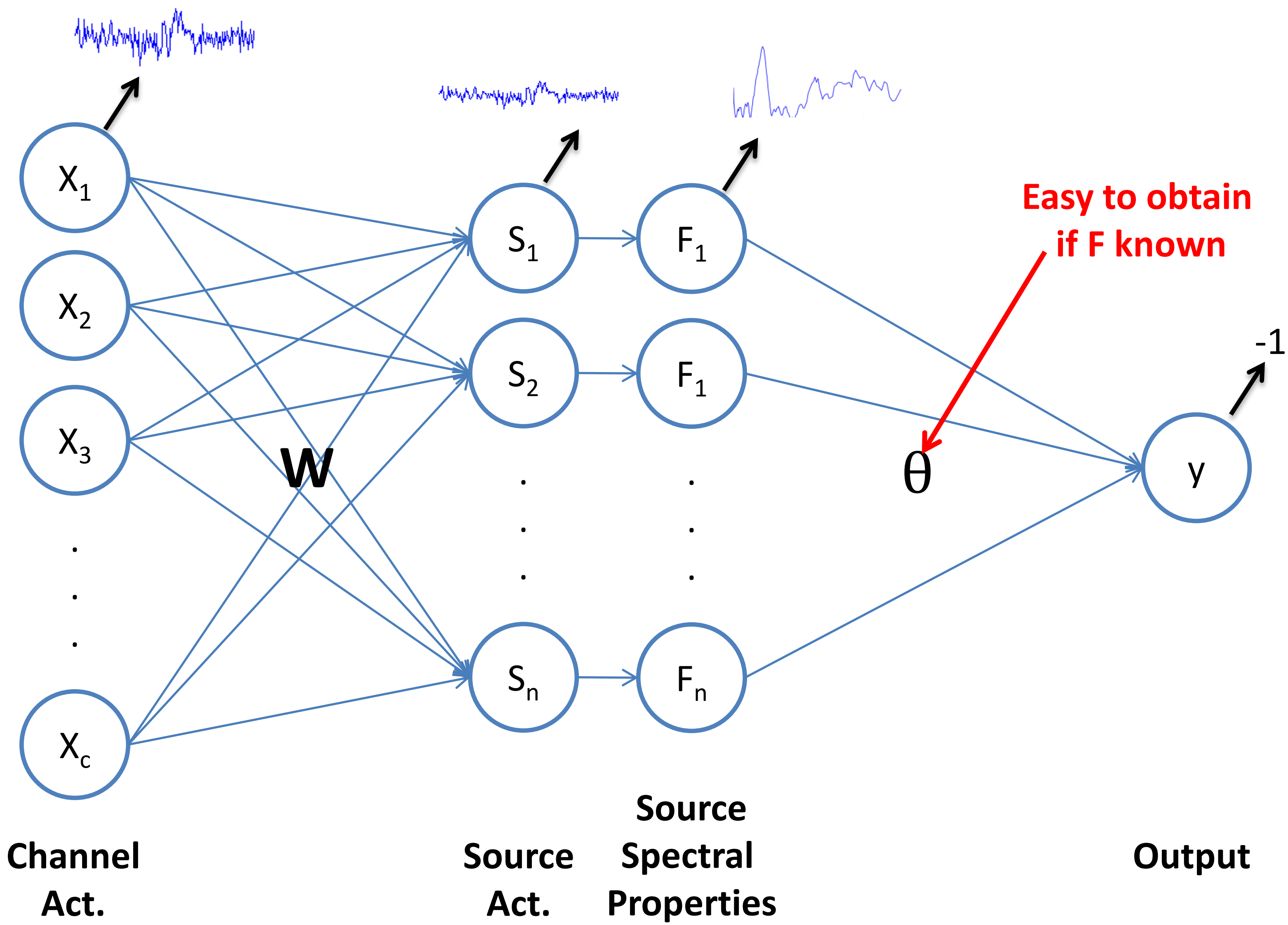
- Nonlinear operation in play, on *source* signals
- Due to, e.g., *shift indeterminacy* of source waveforms
  (no precise time-locking / jitter / high-frequency time course / …)
- Oscillatory processes: e.g., determining the amplitude of source oscillations

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

nonlinear

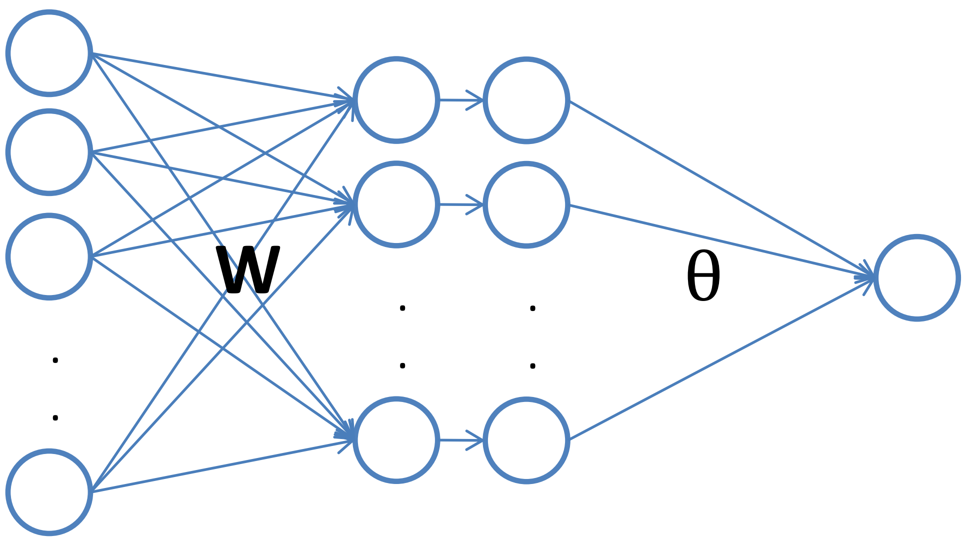
- Nonlinear and discards phase information
  (If done on channels, source spectral properties cannot be recovered)

Latent Variable Viewpoint



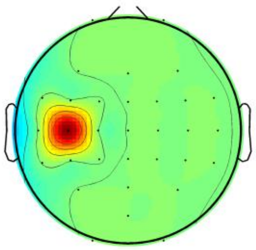
Latent Variable Viewpoint

- 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, …

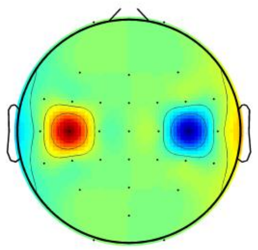


Fixed Filters

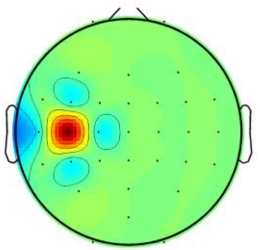
- Simple *a priori* filters (of historical interest)
  - raw channels, common average ref.



  - bipolar derivations



  - surface Laplacian

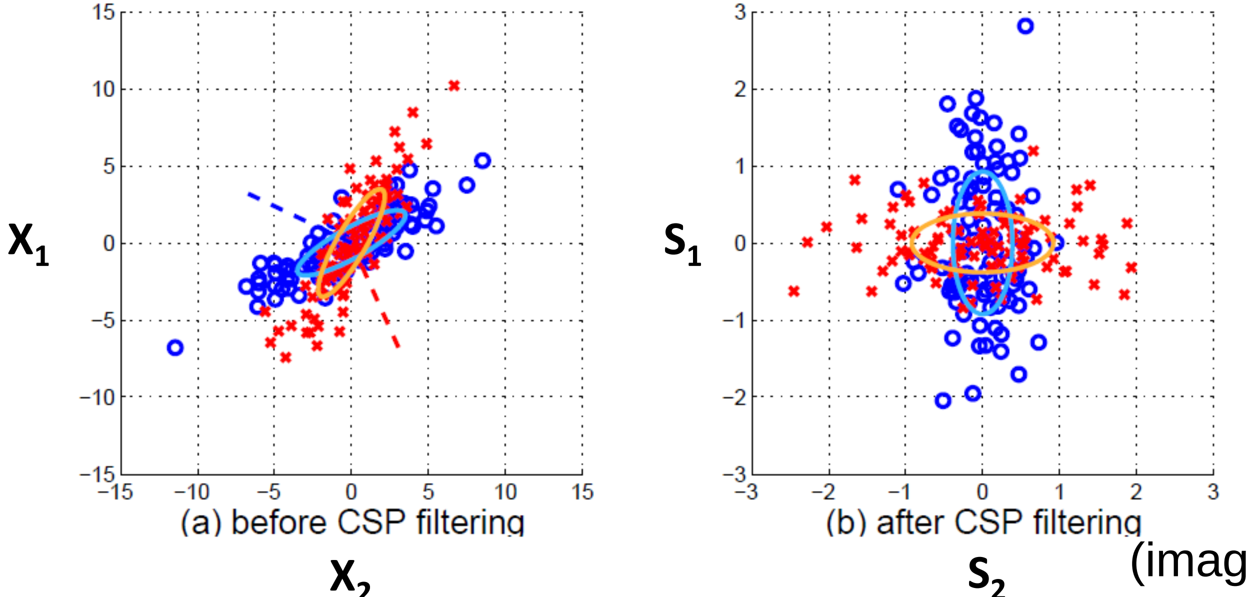


Unsupervised Bottom-Up

- ICA, AMICA
  - **unsupervised**: need to make sure that filters recover the *desired* sources
  - **yields localizable sources**: enables interpretability, enables cortical coregistration, can link to anatomical / functional data (more later)
  - **slow:** problematic between calibration & online use
  - **possible enhancements**: supervised? overcomplete?

Supervised Estimation

- Common Spatial Patterns
  - Most popular algorithm in BCI field
  - Assumption: **Gaussian** Signal, variance features, orthogonal sources (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 $\boldsymbol{\Sigma}_2$

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

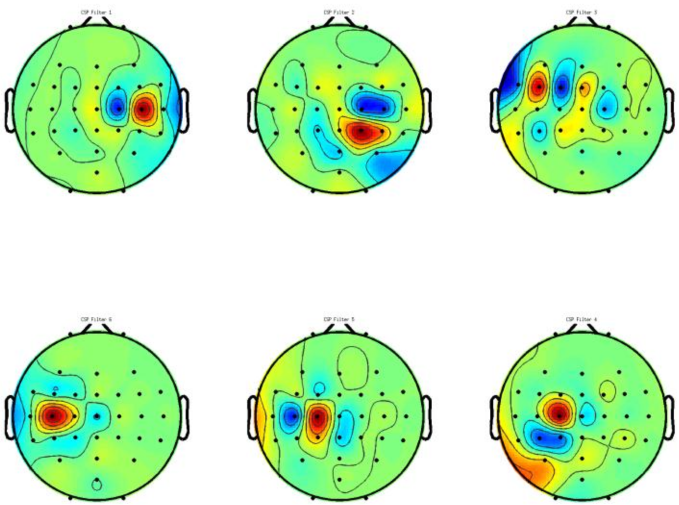
(with $\boldsymbol{\Lambda}_i$ diagonal) such that $\boldsymbol{\Lambda_1} + \boldsymbol{\Lambda_2} = \boldsymbol{I}$. This yields a generalized eigenvalue problem of the form

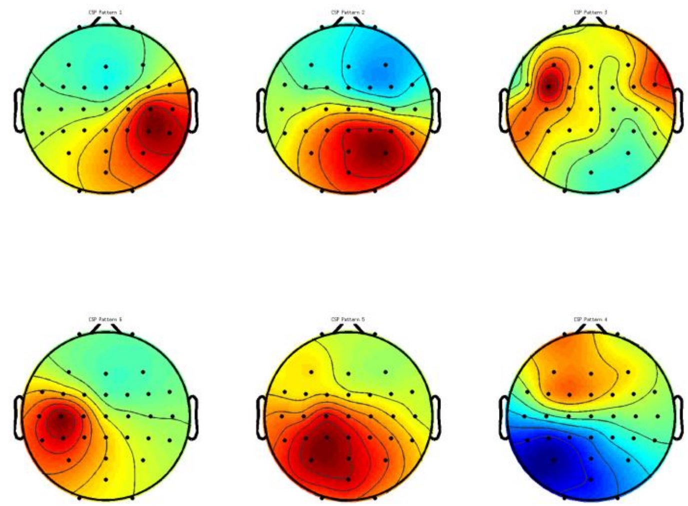
$$\boldsymbol{V}^\top \boldsymbol{\Sigma}_1 \boldsymbol{V} = \boldsymbol{D} \ \wedge \ \boldsymbol{V}^\top (\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2) \boldsymbol{V} = \boldsymbol{I}$$

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)





Supervised Estimation

- Many variations of CSP:
  - Filter-Bank CSP (FBCSP): multiple bands/windows
  - Diagonal Loading CSP (DLCSP): cov. shrinkage
  - Composite CSP (CCSP): covariance prior
  - Tikhonov-regularized CSP (TRCSP): filter shrinkage
  - …
- Complete CSP functional form:

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

Usually learned via LDA

Advanced Supervised Estimation

- Consideration: Given a zero-mean trial $\boldsymbol{X} \in \mathbb{R}^{C\times T}$ with covariance $\boldsymbol{\Sigma} \in \mathbb{R}^{CxC}$, spatial filters $\boldsymbol{W} \in \mathbb{R}^{SxC}$, linear weights $\boldsymbol{\theta} \in \mathbb{R}^{S}$and bias b

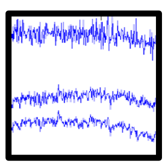
Advanced Supervised Estimation

- Consideration: Given a zero-mean trial $\boldsymbol{X} \in \mathbb{R}^{C \times T}$ with covariance $\boldsymbol{\Sigma} \in \mathbb{R}^{CxC}$, spatial filters $\boldsymbol{W} \in \mathbb{R}^{SxC}$, linear weights $\boldsymbol{\theta} \in \mathbb{R}^{S}$and bias b

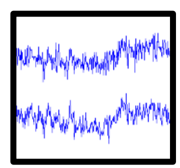
- Omitting the log from CSP, we have:

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

X



WX



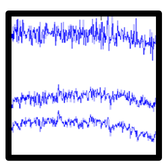
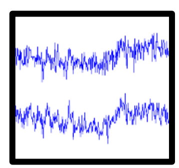
var(**WX**) $\begin{bmatrix} 2.3 \\ 4.5 \end{bmatrix}$

Advanced Supervised Estimation

- Consideration: Given a zero-mean trial $\boldsymbol{X} \in \mathbb{R}^{C \times T}$ with covariance $\boldsymbol{\Sigma} \in \mathbb{R}^{CxC}$, spatial filters $\boldsymbol{W} \in \mathbb{R}^{SxC}$, linear weights $\boldsymbol{\theta} \in \mathbb{R}^{S}$ and bias b

- Omitting the log from CSP, we have:

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

X  **WX**  var(**WX**) $\begin{bmatrix} 2.3 \\ 4.5 \end{bmatrix}$

- Rewriting in terms of individual spatial filters $\boldsymbol{W}_k$:

$$y = b + \sum_{k=1}^{S} \boldsymbol{\theta}_k \, \mathrm{var}(\boldsymbol{W}_k\boldsymbol{X})$$

Advanced Supervised Estimation

- The variance term can be expressed using the covariance matrix $\boldsymbol{\Sigma}$ of segment $\boldsymbol{X}$:

$$y = b + \sum_{k=1}^{S} \boldsymbol{\theta}_k \,\mathrm{var}(\boldsymbol{W}_k \boldsymbol{X}) = b + \sum_{k=1}^{S} \boldsymbol{\theta}_k \left(\boldsymbol{W}_k \boldsymbol{\Sigma} \boldsymbol{W}_k{}^{\top}\right)$$

Advanced Supervised Estimation

- The variance term can be expressed using the covariance matrix $\boldsymbol{\Sigma}$ of segment $\boldsymbol{X}$:

$$y = b + \sum_{k=1}^{S} \boldsymbol{\theta}_k \operatorname{var}(\boldsymbol{W}_k \boldsymbol{X}) = b + \sum_{k=1}^{S} \boldsymbol{\theta}_k \left(\boldsymbol{W}_k \boldsymbol{\Sigma} \boldsymbol{W}_k{}^{\top}\right)$$

- And $\boldsymbol{W}_k \boldsymbol{\Sigma} \boldsymbol{W}_k{}^{\top}$ can be replaced by the inner product between two matrices $\left\langle \boldsymbol{W}_k \boldsymbol{W}_k{}^{\top}, \boldsymbol{\Sigma} \right\rangle$

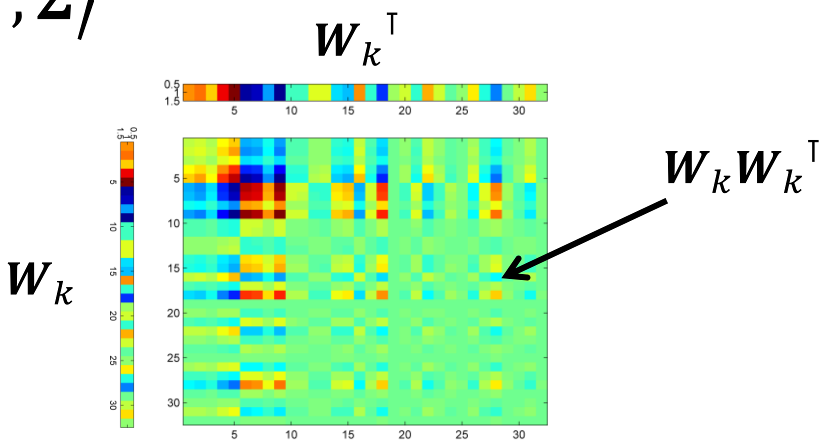
Advanced Supervised Estimation

- The variance term can be expressed using the covariance matrix $\boldsymbol{\Sigma}$ of segment $\boldsymbol{X}$:

$$y = b + \sum_{k=1}^{S} \boldsymbol{\theta}_k \operatorname{var}(\boldsymbol{W}_k \boldsymbol{X}) = b + \sum_{k=1}^{S} \boldsymbol{\theta}_k \left(\boldsymbol{W}_k \boldsymbol{\Sigma} \boldsymbol{W}_k^{\top}\right)$$

- And $\boldsymbol{W}_k \boldsymbol{\Sigma} \boldsymbol{W}_k^{\top}$ can be replaced by the inner product between two matrices $\langle \boldsymbol{W}_k \boldsymbol{W}_k^{\top}, \boldsymbol{\Sigma} \rangle$

$$b + \sum_{k=1}^{S} \boldsymbol{\theta}_k \langle \boldsymbol{W}_k \boldsymbol{W}_k^{\top}, \boldsymbol{\Sigma} \rangle$$



Advanced Supervised Estimation

- The variance term can be expressed using the covariance matrix $\boldsymbol{\Sigma}$ of segment $\boldsymbol{X}$:

$$y = b + \sum_{k=1}^{S} \boldsymbol{\theta}_k \operatorname{var}(\boldsymbol{W}_k \boldsymbol{X}) = b + \sum_{k=1}^{S} \boldsymbol{\theta}_k \left(\boldsymbol{W}_k \boldsymbol{\Sigma} \boldsymbol{W}_k^{\top}\right)$$

- And $\boldsymbol{W}_k \boldsymbol{\Sigma} \boldsymbol{W}_k^{\top}$ can be replaced by the inner product between two matrices $\langle \boldsymbol{W}_k \boldsymbol{W}_k^{\top}, \boldsymbol{\Sigma} \rangle$, and regrouped:

$$b + \sum_{k=1}^{S} \boldsymbol{\theta}_k \langle \boldsymbol{W}_k \boldsymbol{W}_k^{\top}, \boldsymbol{\Sigma} \rangle = b + \left\langle \sum_{k=1}^{S} \boldsymbol{\theta}_k \boldsymbol{W}_k \boldsymbol{W}_k^{\top}, \boldsymbol{\Sigma} \right\rangle$$

$$= b + \langle \boldsymbol{\Theta}, \boldsymbol{\Sigma} \rangle$$

Advanced Supervised Estimation

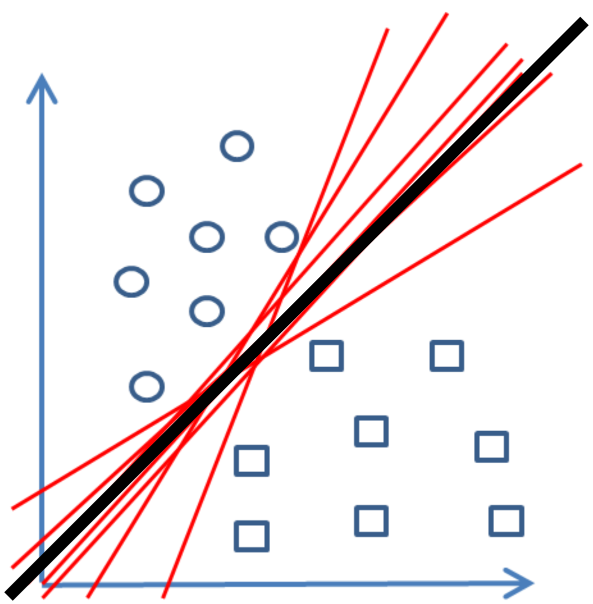
- Thus this form is linear in the *covariance matrix* of X:

$$y = b + \langle \boldsymbol{\Theta}, \boldsymbol{\Sigma} \rangle = \boldsymbol{b} + \widetilde{\boldsymbol{\theta}}\mathrm{vec}(\boldsymbol{\Sigma})$$

- Could again learn $\widetilde{\boldsymbol{\theta}}$ using a simple linear method (e.g., LDA), but *very* high-dimensional (#parameters=${C^2}^2$)
- Need a method suitable for large-scale problems

Large-Scale Models

- *Discriminative* learning approaches like Support Vector Machines (SVMs) and Generalized Linear Models (GLMs) are well-adapted to high-dimensional / large-scale problems
- These directly optimize the parameters $\boldsymbol{\theta}$ given the data

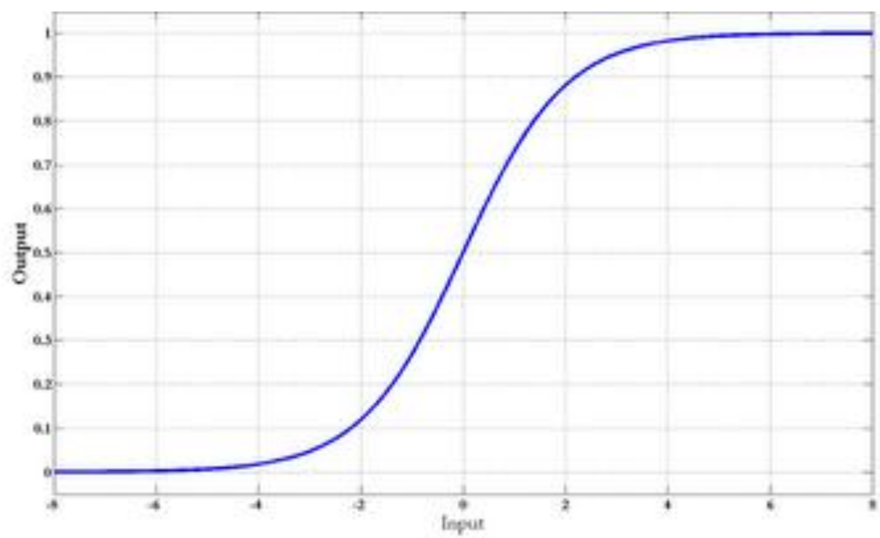


Swartz Center for Computational Neuroscience

Large-Scale Models

- Logistic Regression is a GLM that maps onto binary outputs via a logistic "link function"

$$q_\theta(Y = y|X) = \frac{1}{1 + e^{-y f_\theta(\boldsymbol{X})}}, (y \in \{-1, +1\})$$



$-\ f_0(\boldsymbol{X}) \rightarrow$

Large-Scale Models

- Logistic Regression is a GLM that maps onto binary outputs via a logistic "link function"

$$q_\theta(Y = y|X) = \frac{1}{1 + e^{-y f_\theta(\boldsymbol{X})}}, (y \ \in \{-1, +1\})$$

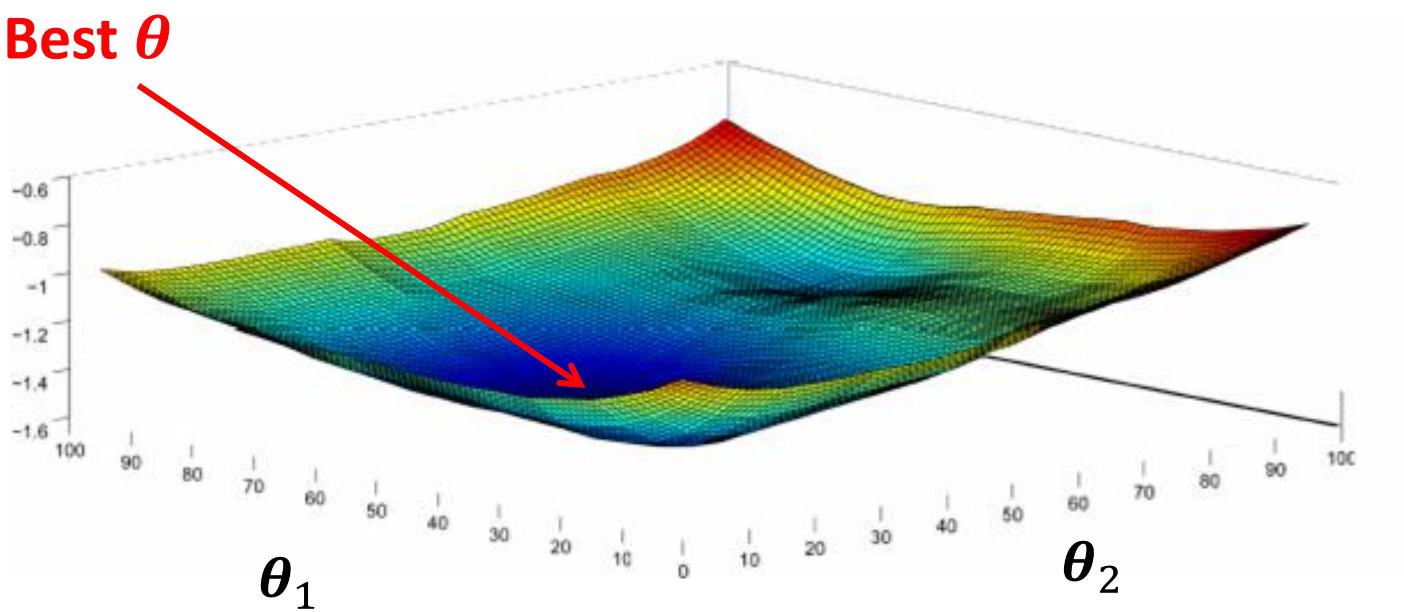
- … and linear function $f_\theta(\boldsymbol{X})$

$$f_\theta(\boldsymbol{X}) = \ \boldsymbol{\theta X} + b$$

Large-Scale Models

- Trick: $\boldsymbol{\theta}$ can be obtained via off-the-shelf *convex optimization* methods (such as CVX) by solving the problem

$$\min_{\boldsymbol{\theta}} \log\left(1 + e^{-\boldsymbol{y} f_{\theta}(\boldsymbol{X})}\right)$$



Large-scale Models

- For large problems, solution is still prone to over-fitting – need to plug in *additional assumptions*

$$\min_{\boldsymbol{\theta}} \log\left(1 + e^{-y f_\theta(\boldsymbol{X})}\right) + \lambda\Omega(\boldsymbol{\theta})$$

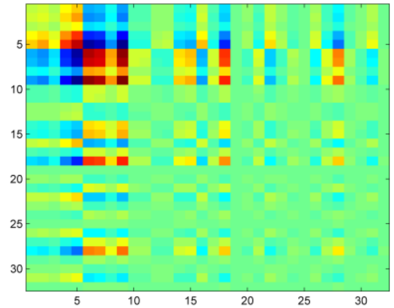
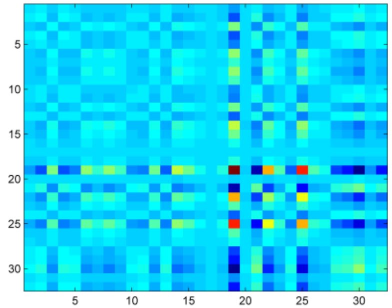
- Many choices for regularization term Ω
  - $\Omega(\boldsymbol{\theta}) = \|\boldsymbol{\theta}\|_2$ encourages small weights
  - $\Omega(\boldsymbol{\theta}) = \|\boldsymbol{\theta}\|_1 = |\boldsymbol{\theta}_1| + |\boldsymbol{\theta}_2| + \cdots$ encourages *sparsity*
  - can also get sparsity on groups of weights
  - combinations thereof, …

Large-Scale Learning Applied

- In the previous supervised oscillatory model $y = b + \langle \boldsymbol{\Theta}, \boldsymbol{\Sigma} \rangle$, the matrix-shaped $\boldsymbol{\Theta}$ allows for a special matrix norm regularization:

$$\min_{\boldsymbol{\Theta}} \log\left(1 + e^{-y f_{\boldsymbol{\Theta}}(\boldsymbol{X})}\right) + \lambda \sum_{k=1}^{rank(\boldsymbol{\Theta})} \sigma_k(\boldsymbol{\Theta})$$

- This encourages a low-rank structure in $\boldsymbol{\Theta}$, i.e.

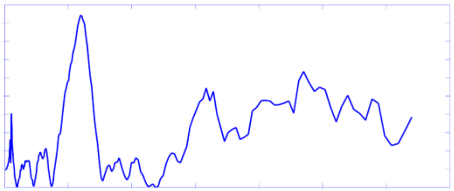
$\boldsymbol{\Theta}$ =  +  + ...

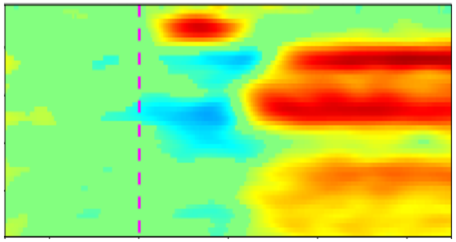
Back to ICA

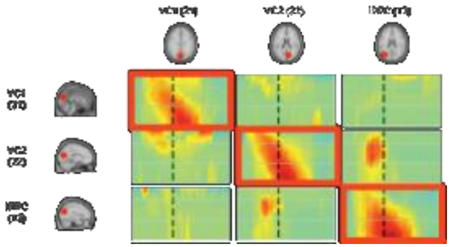
- ICA can learn spatial filters W explicitly, yields meaningful source activations S
- Can use any spectral measure on trial segments of S to extract oscillatory structure
- Can learn relationship between oscillatory structure and cognitive state using simple or complex approaches…

Some Spectral Measures

- Band-power
  - Band-pass (e.g., FIR, IIR, ...) + (log-)variance
- Fourier spectrum
  - Windowed DFT/FFT (e.g., Hann)
  - Welch method
  - Multi-taper method
- Time/Frequency representations
  - Short-Time Fourier Transform (STFT)
  - Continuous Wavelet Transform (CWT)
  - Discrete wavelet transform (DWT)
- Coherence, Effective Connectivity, ...

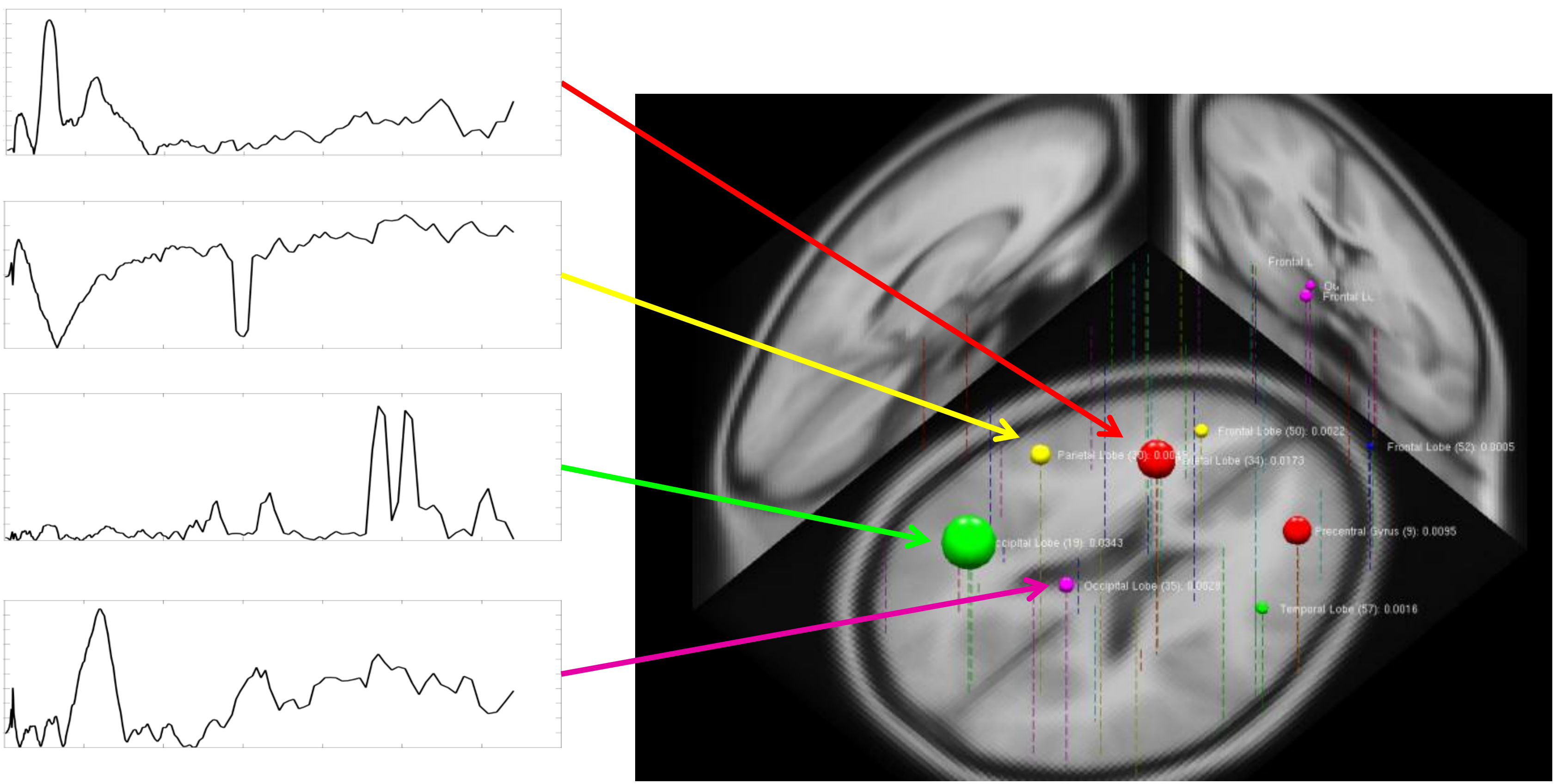






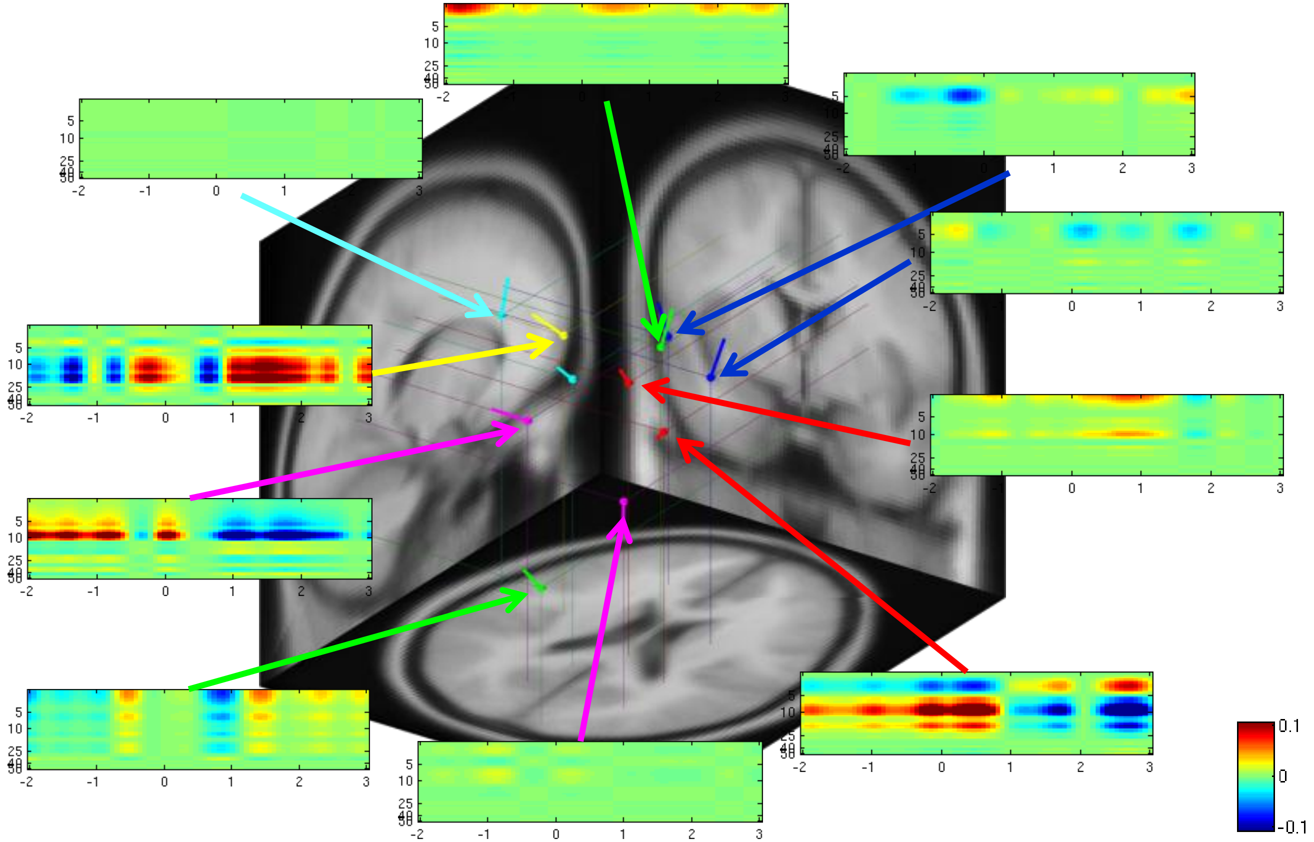
Source-Space Modeling

- If IC sources are localized using, e.g., dipole fitting or NFT, parameters ($\boldsymbol{\theta}$) have a location



Source-Space Modeling

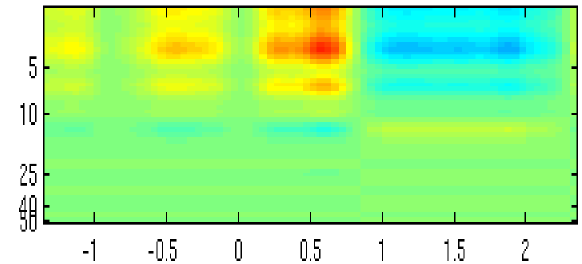
- If IC sources are localized using, e.g., dipole fitting or NFT, parameters ($\boldsymbol{\theta}$) have a location

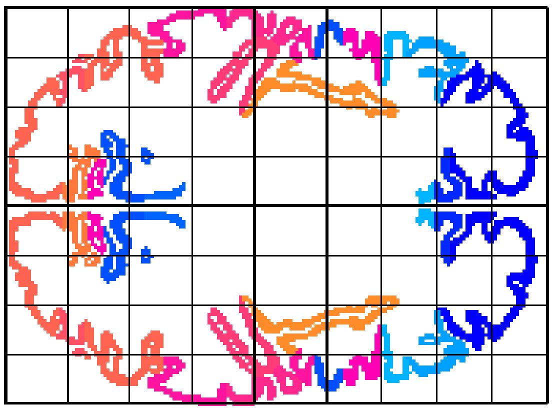


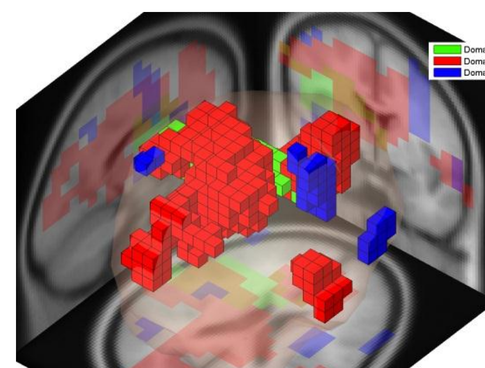
Swartz Center for Computational Neuroscience

Source-Space Modeling

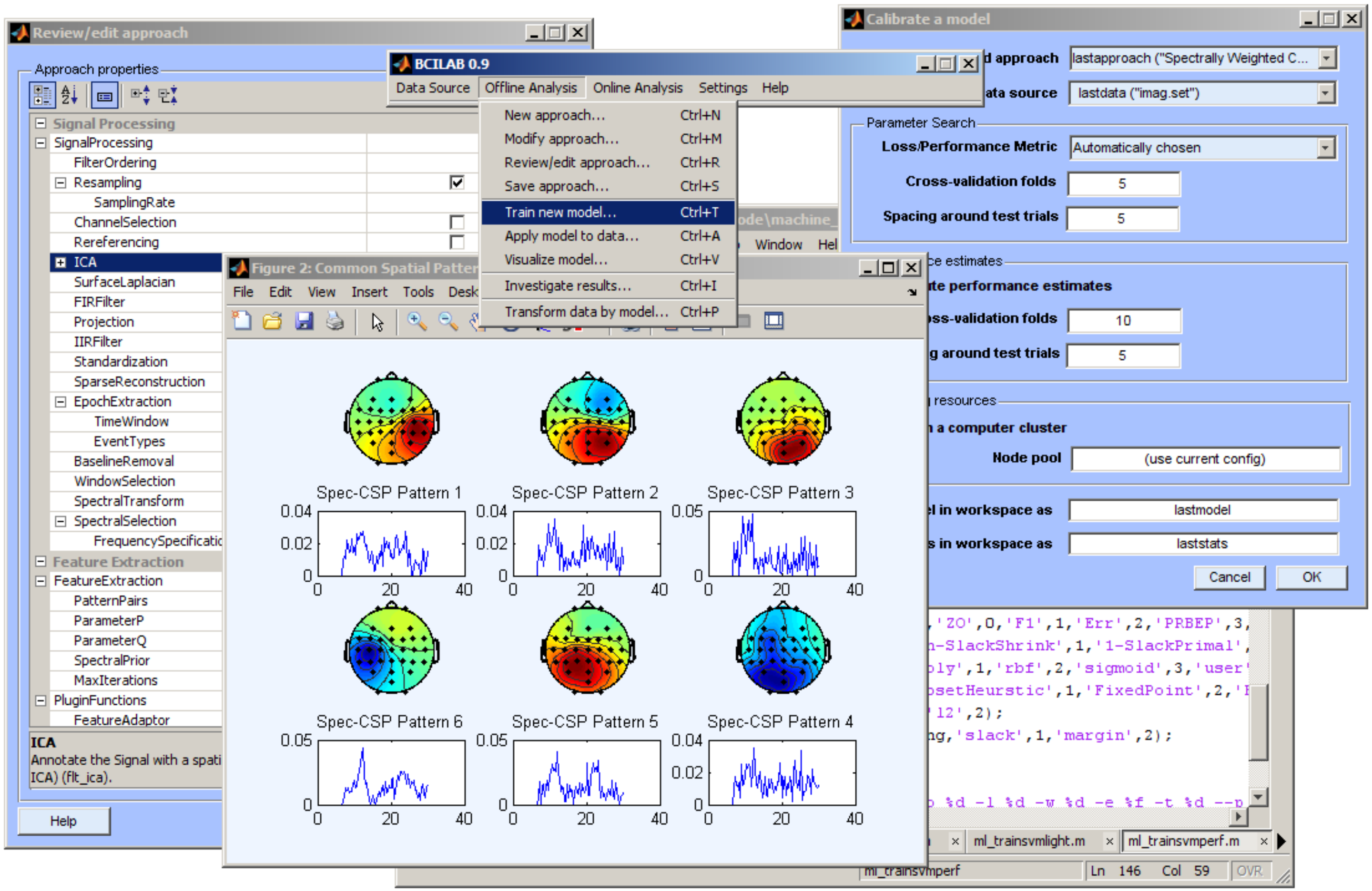
- Structural prior knowledge
  - can be introduced as side assumptions in the model (e.g. smoothness, sparsity, group sparsity, low rank, …)
- Quantitative prior knowledge
  - Structure atlases (Talairach, LONI, …) can supply information about the *a priori* relevance of a brain process
  - Can adapt the per-parameter penalty
- Empirical data
  - Data collected from other subjects can be co-registered/aligned and yield empirical prior distributions







Next: BCILAB Practicum

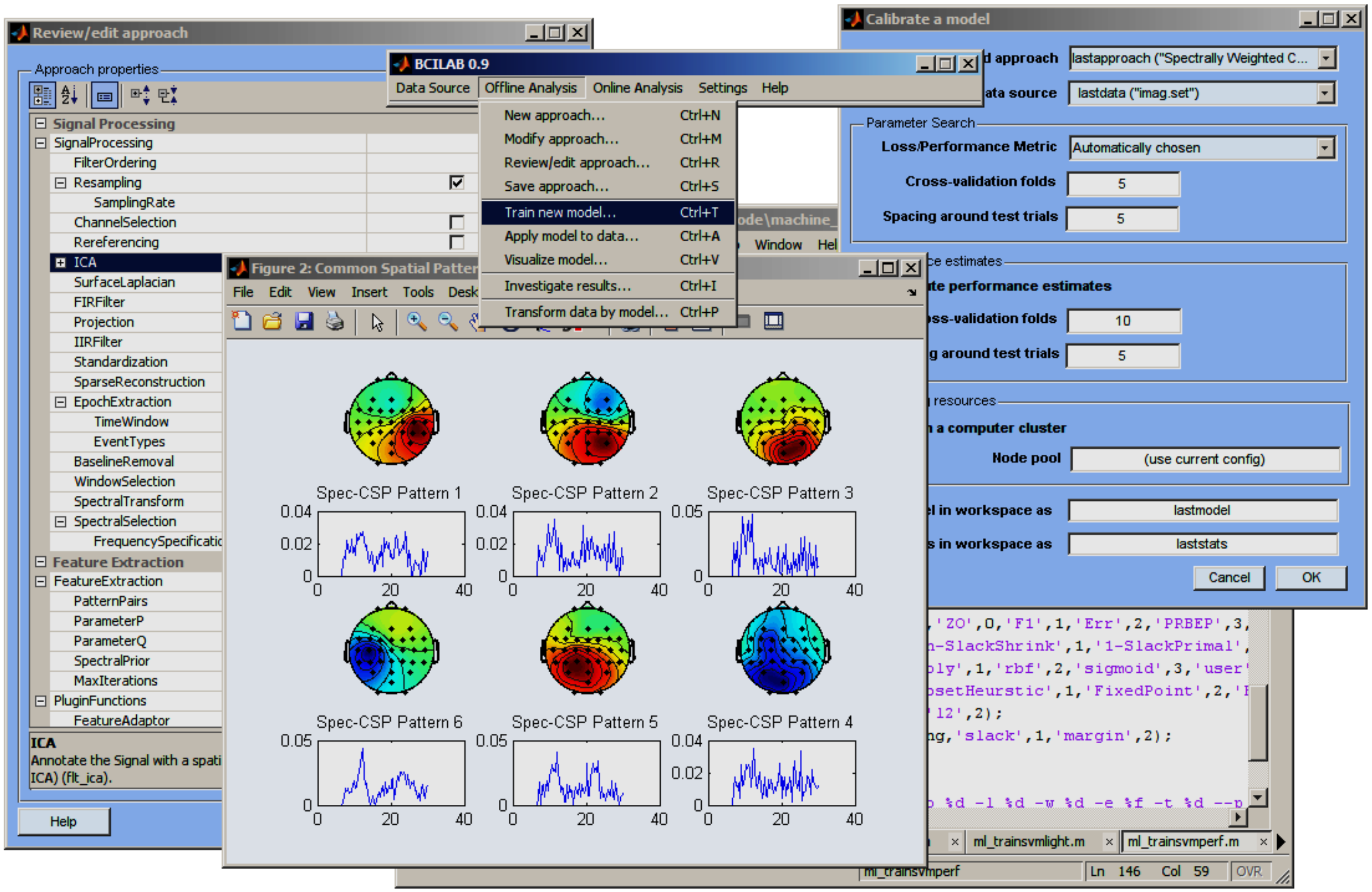


[http://sccn.ucsd.edu/**wiki/BCILAB**](http://sccn.ucsd.edu/wiki/BCILAB)

Thanks!

Questions?

BCILAB Briefing



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

Idea & Purpose

- Like EEGLAB, but for BCI (and/or cognitive state assessment)
  - Seeding a community
  - Strengthening links between BCI and Neuroscience
- SCCN's in-house tool for BCI problems
  - Main focus: Advanced cognitive monitoring
  - Part of a large US research program (CaN CTA)
  - Funded by ARL (and ONR, Swartz Foundation, …)

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, …)
- Next-generation framework
  - Fully probabilistic
  - Model inference from data corpora*
  - Anatomical priors, other neuroscience-aware features
  - Processing of parallel streams

(*: not yet in the current release)

BCILAB Components

Framework

GUI / Scripting Interfaces

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

Plugins

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

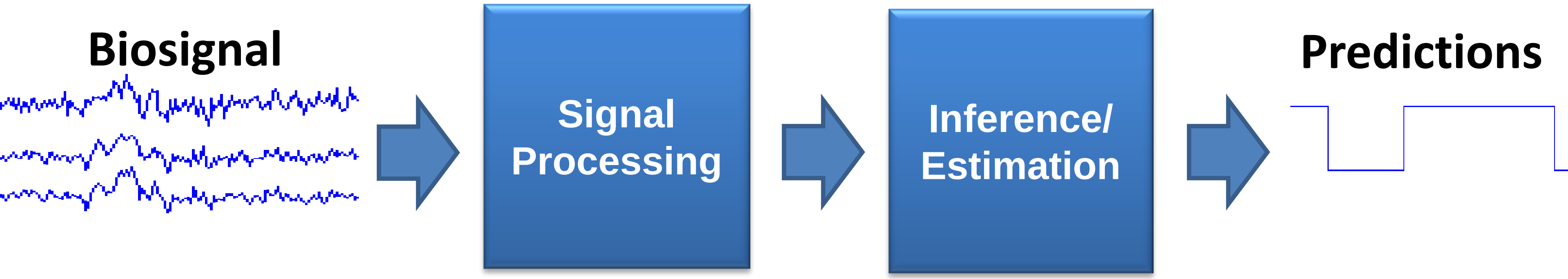
CVX | BNT | EEGLAB | GUI utils | LIBSVM | GLMNET | … | Driver I/O

Signal Processing?

- Some signal-level computations can be done more efficiently than window-by-window (esp. when successive windows overlap a lot)

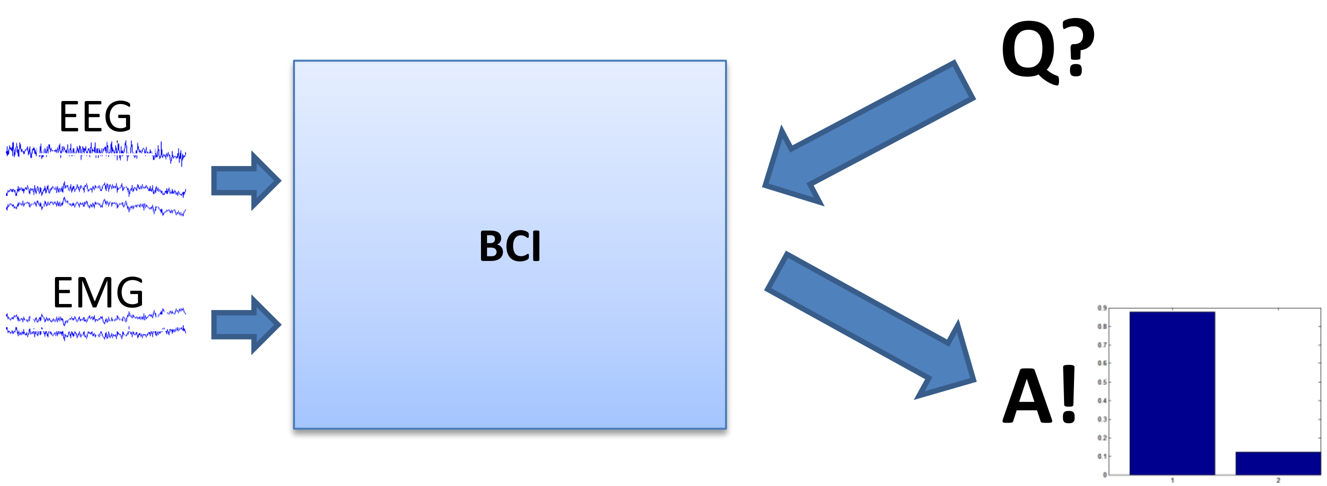
Signal Processing?

- Some signal-level computations can be done more efficiently than window-by-window (esp. when successive windows overlap a lot)
- Room for good DSP use (e.g., frequency filter, spatial filter, …) before actual prediction
- Also, can assemble approaches from existing components



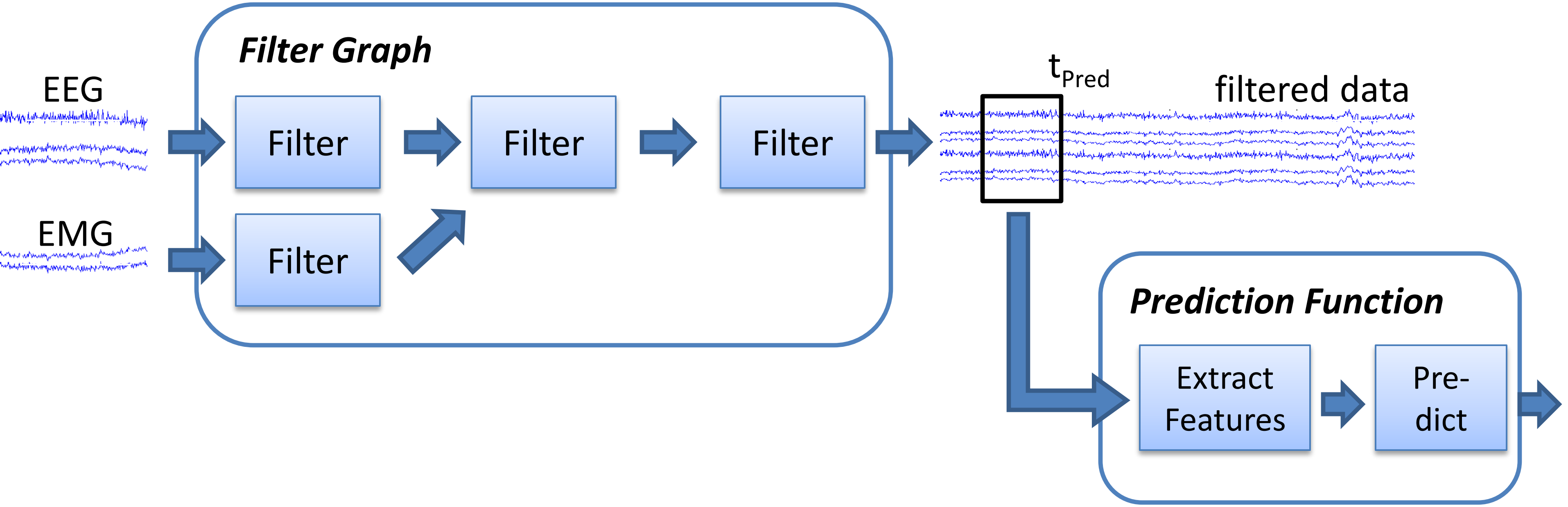
BCI Behavior

- BCIs in BCILAB are acting as an oracle that consumes one or more biosignals and can respond to (pre-defined) queries about cognitive state



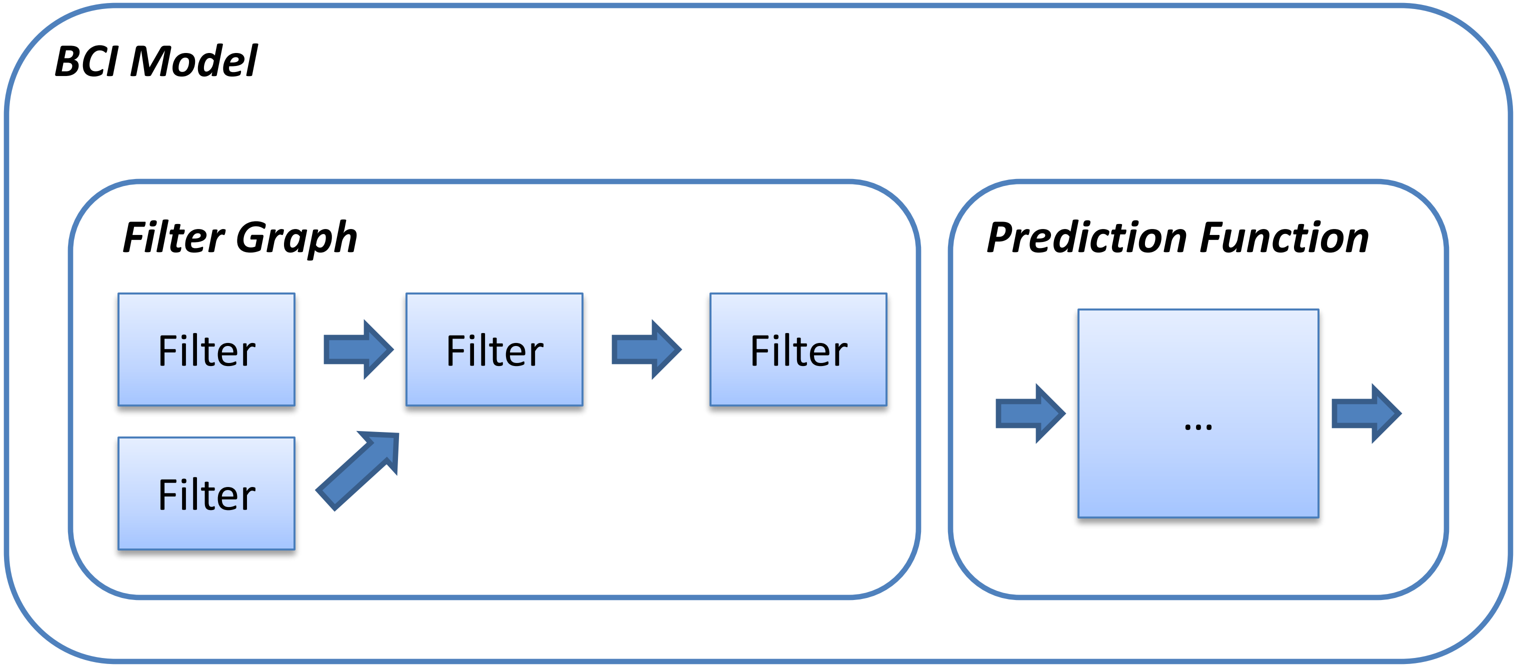
Online Data Flow

- A filter graph receives all input samples and produces pre-filtered data
- The prediction function may be queried on demand on the filter graph's outputs



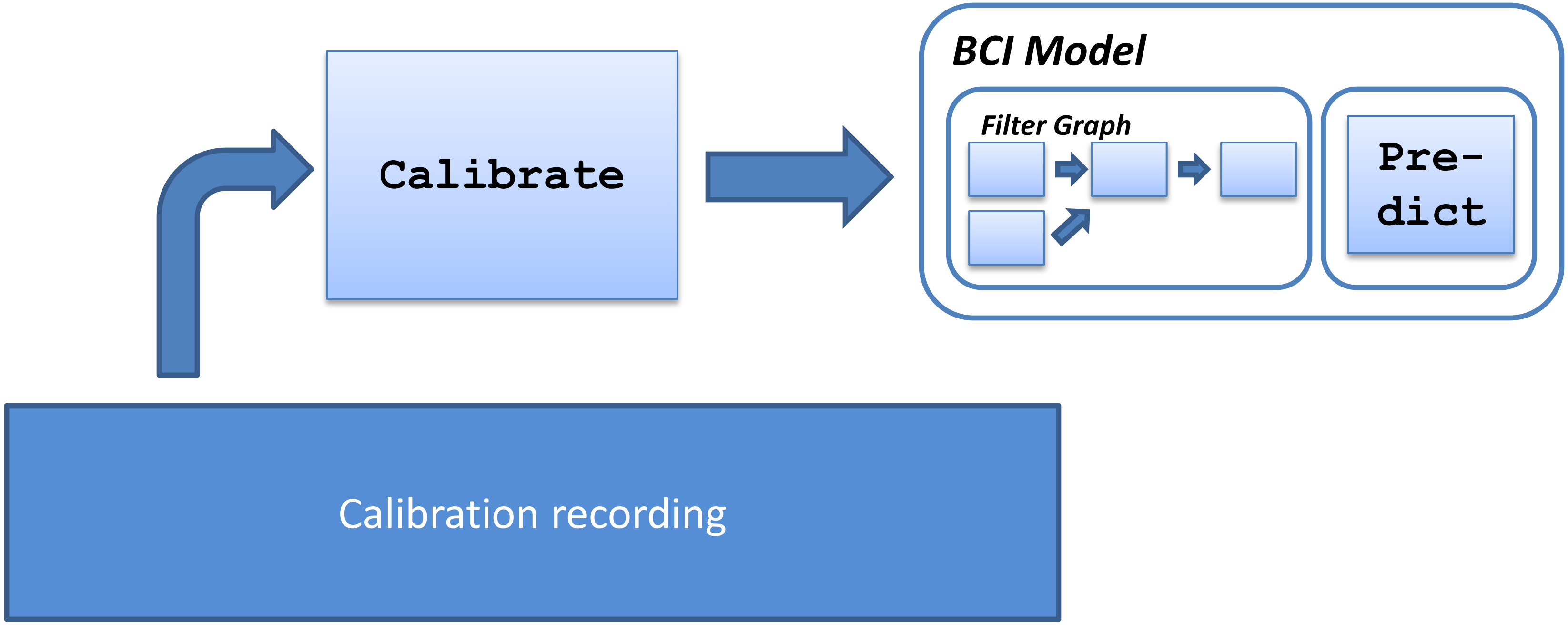
BCI Models

- BCIs are described by "BCI models" that specify both the *filter graph* and the *prediction function* (incl. parameters)



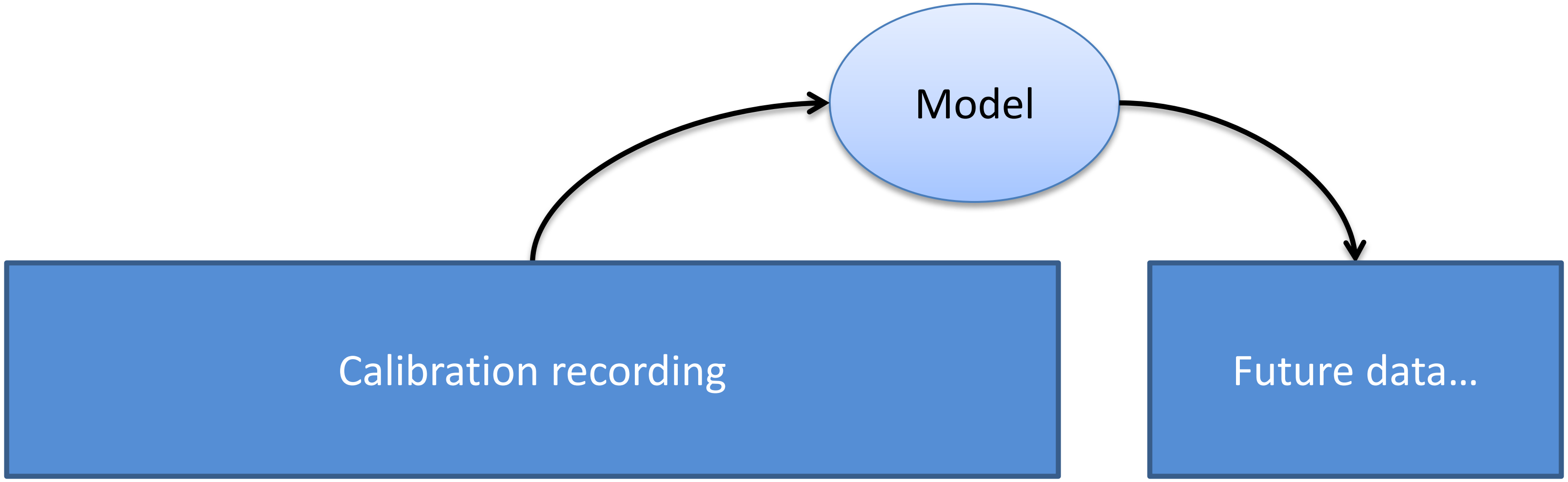
BCI Paradigms

- BCI paradigms are the coarsest plugin type in BCILAB and tie all parts of a BCI approach together
- They are seeds for new BCI designs and cornerstones of BCILAB usage



Offline Evaluation

- 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)



Offline Evaluation

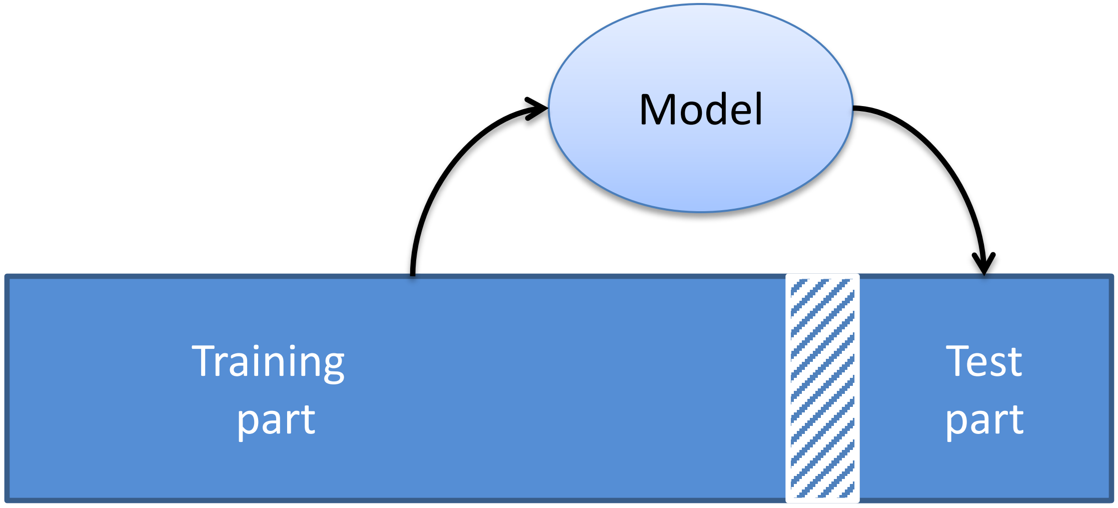
- Evaluation of computational approaches on a **single** data set?

?

Calibration recording

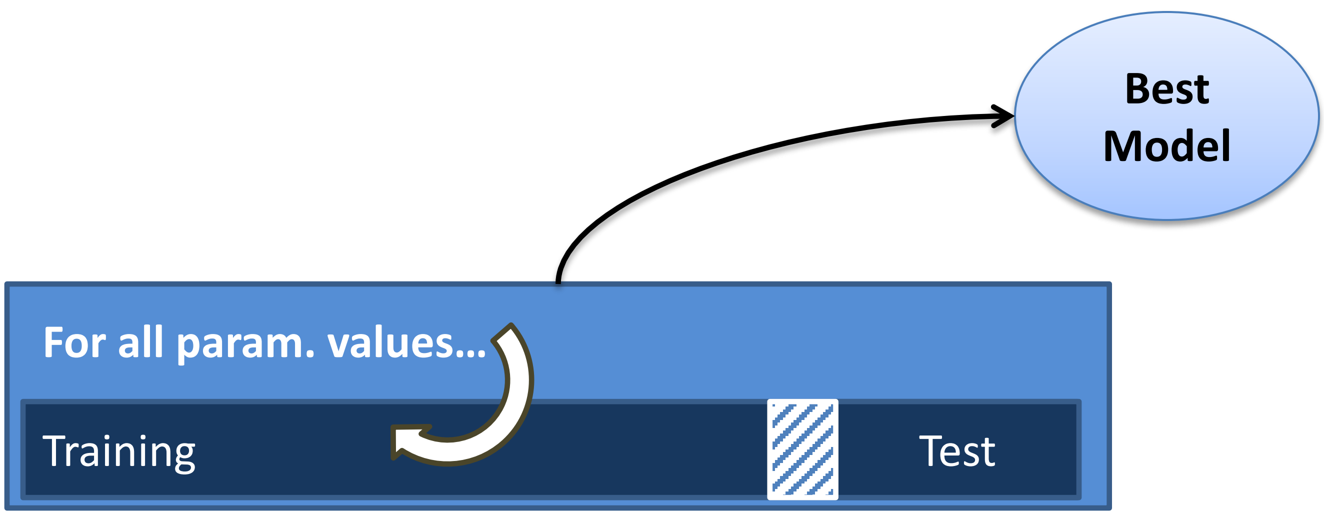
Offline Evaluation

- 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*



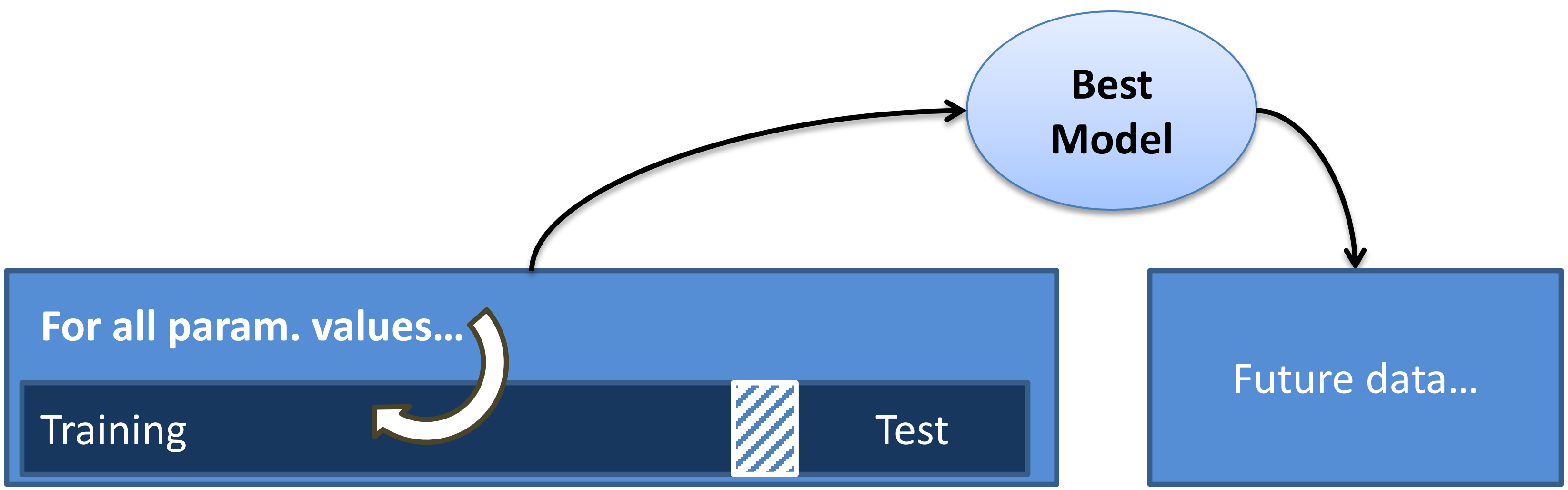
Resolving Free Parameters

- 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")