Lecture 7: Oscillatory Processes

Introduction to Modern Brain-Computer Interface Design

Christian A. Kothe
SCCN, UCSD

Outline

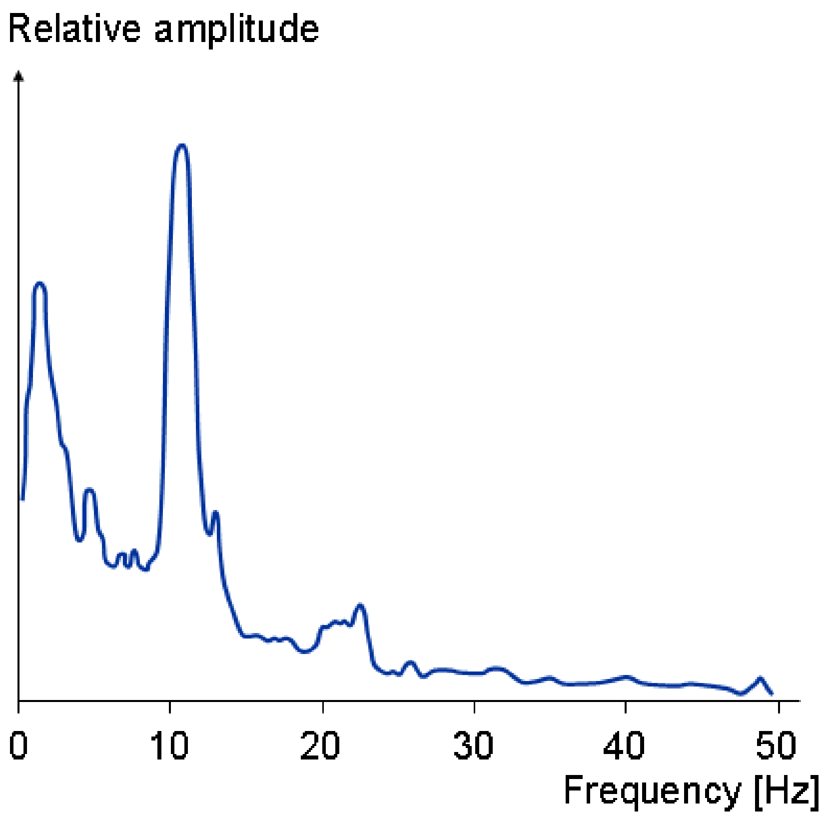
1. Basics and Examples
2. The Spatial Filter Problem
3. Common Spatial Patterns
4. Alternatives and Extensions

Swartz
Center for
Computational
Neuroscience

7.1 Basics and Examples

Oscillatory Processes

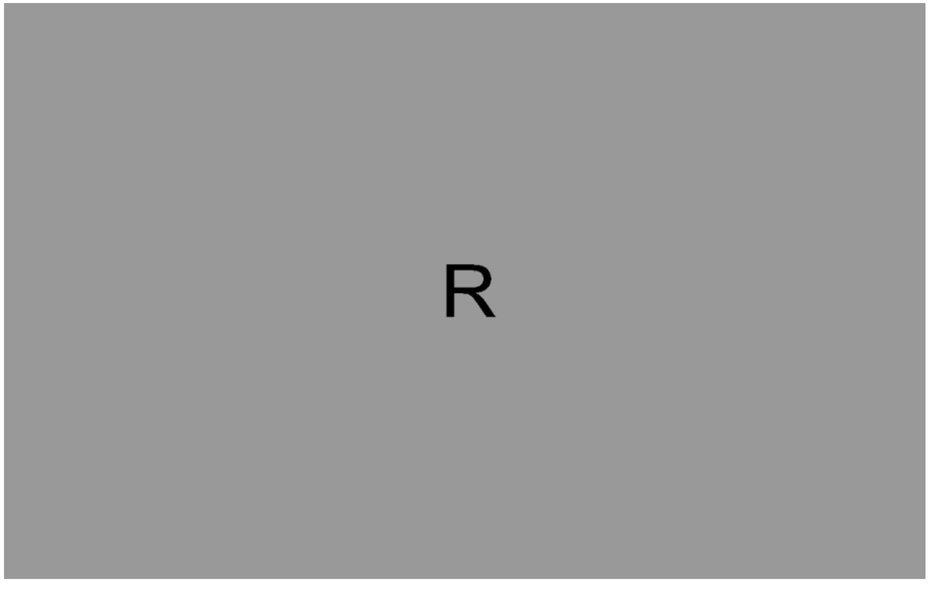
- **Best example:** cortical idle rhythms, e.g. occipital alpha, motor cortex alpha+beta



Malmivuo and Plonsey, 1995

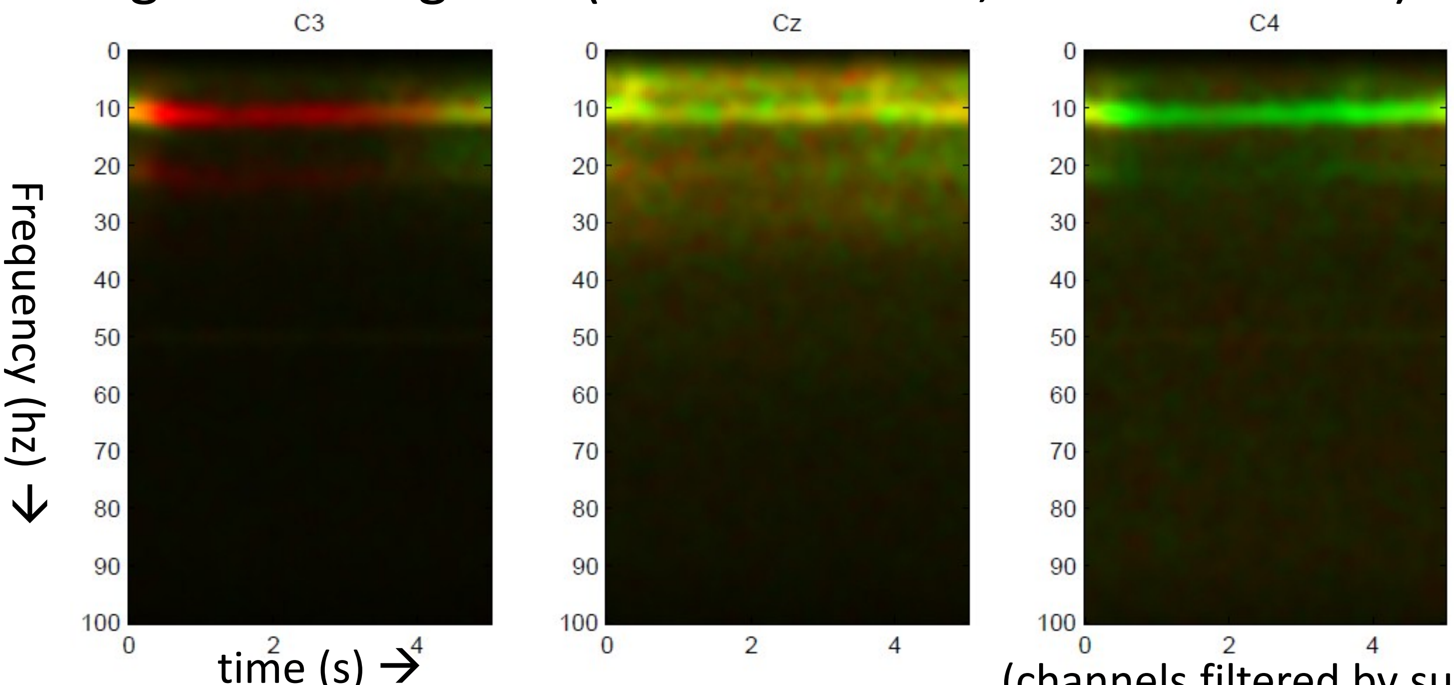
Experimental Task

- The experiment consists of 160 trials (pause at ½ the experiment) . Each trial begins with a letter (either L or R) displayed for 3s. The subject is instructed to subsequently imagine either a left-hand or a right-hand movement. Each trial ends with a blank screen displayed for 3.5s.



Motor Cortex ERD/ERS

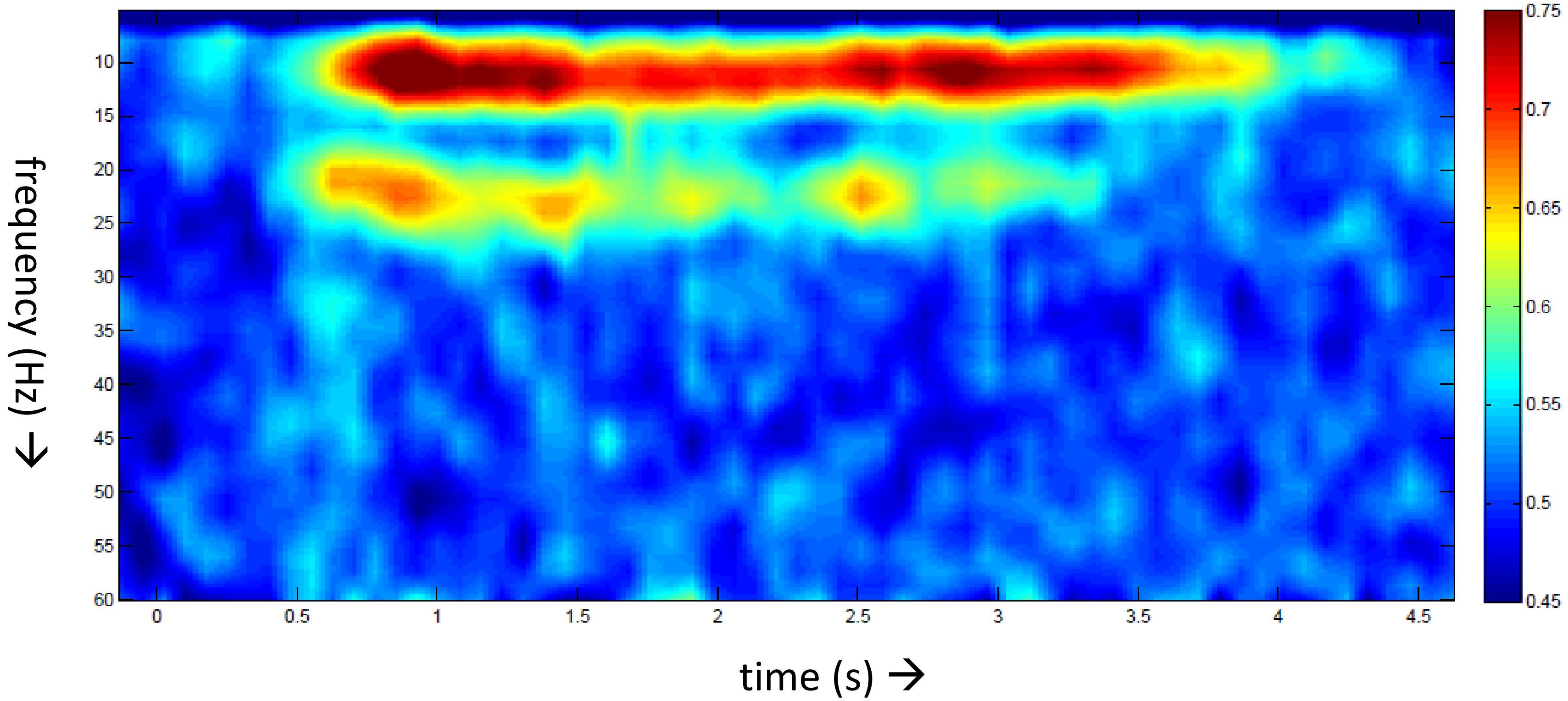
- **Event-Related Synchronization / Desynchronization:** attentuation of motoric idle rhythms in response to an event
- Average spectrogram for left-hand movement imagination in red **+** average spectrogram for right-hand movement imagination in green (160 trials each, stimulus at t=0)



(channels filtered by surface Laplacian)

Motor Cortex ERD/ERS

- Alternative visualization of information content per time-frequency resel, same data:



7.2 The Spatial Filter Problem

Quantifying Oscillatory Processes

- Nonlinear operation in play, on *source* signals
- Necessary due to *shift indeterminacy* of source waveforms (no precise time/phase-locking, jitter …)
- In oscillatory processes represented by determining the amplitude of source oscillations

$$\boldsymbol{S} = \boldsymbol{W}\boldsymbol{X} \qquad \boldsymbol{F} = \|\mathrm{DFT}(\boldsymbol{S})\| \qquad y = \boldsymbol{\theta}\boldsymbol{F} + b$$

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

Quantifying Oscillatory Processes

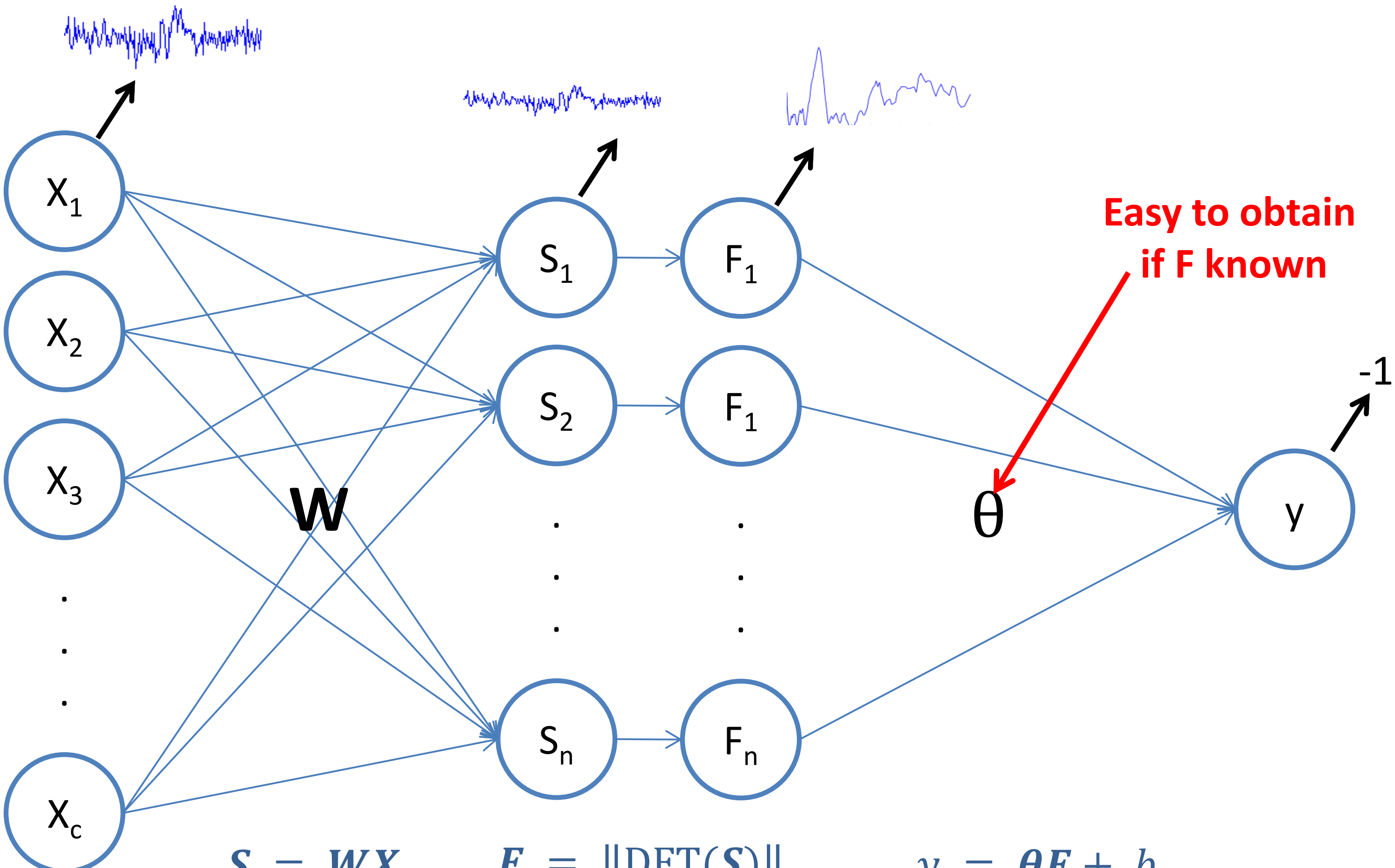
- Nonlinear operation in play, on *source* signals
- Necessary due to *shift indeterminacy* of source waveforms (no precise time/phase-locking, jitter …)
- In oscillatory processes represented by determining the amplitude of source oscillations

$$\boldsymbol{S} = \boldsymbol{W}\boldsymbol{X} \qquad \boldsymbol{F} = \|\mathrm{DFT}(\boldsymbol{S})\| \qquad y = \boldsymbol{\theta}\boldsymbol{F} + b$$

culprit

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

Latent Variable Viewpoint

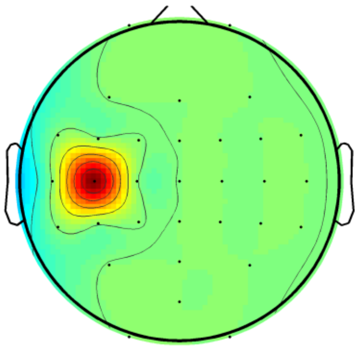


$S = WX$ $\quad F = \|\text{DFT}(S)\|$ $\quad y = \theta F + b$

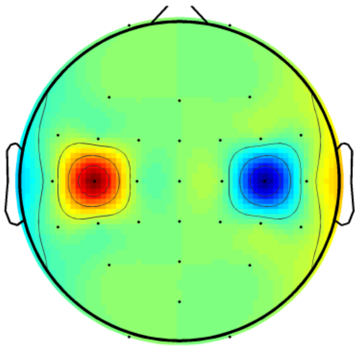
Channel Activity | Source Activity | Source Spectrum | Output

How to Learn the Spatial Filter W?

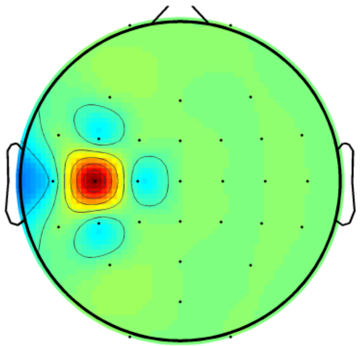
- **Option A – No Learning**: use fixed *ad hoc* filters instead
- Performance not abysmal, but *far from optimal* – room for improvement



Common Average Reference



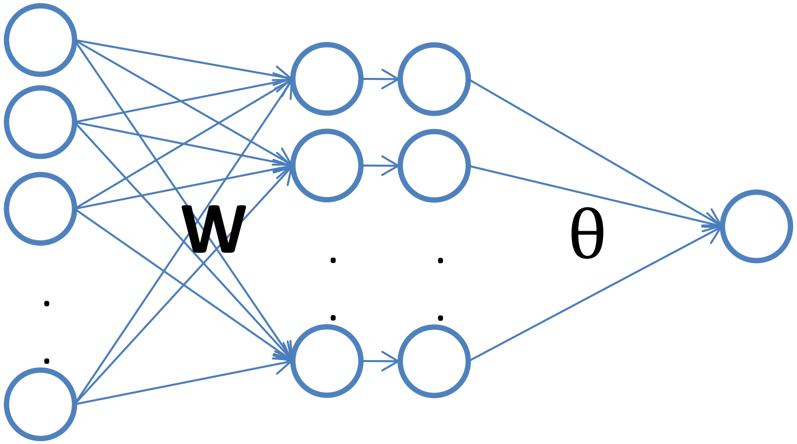
Bipolar Derivations



Surface Laplacian Derivations

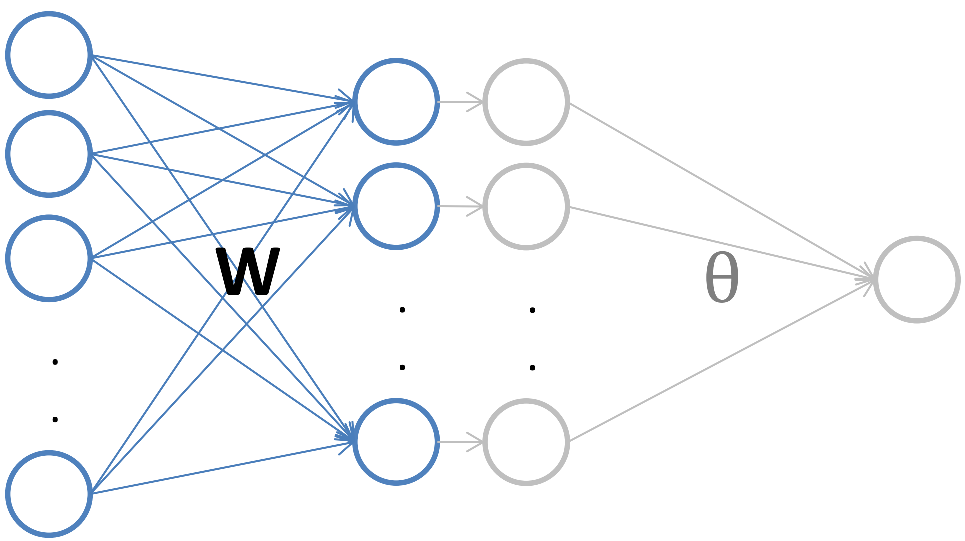
How to Learn the Spatial Filter W?

- **Option B – Top-down**: using neural-network like back-propagation / gradient descent (supervised learning)
- Inputs **X** are known, desired outputs y are known, spectral mapping in between is known
- For any ($\mathbf{W}$,$\boldsymbol{\theta}$) can calculate the loss given known **X** and y, and update it



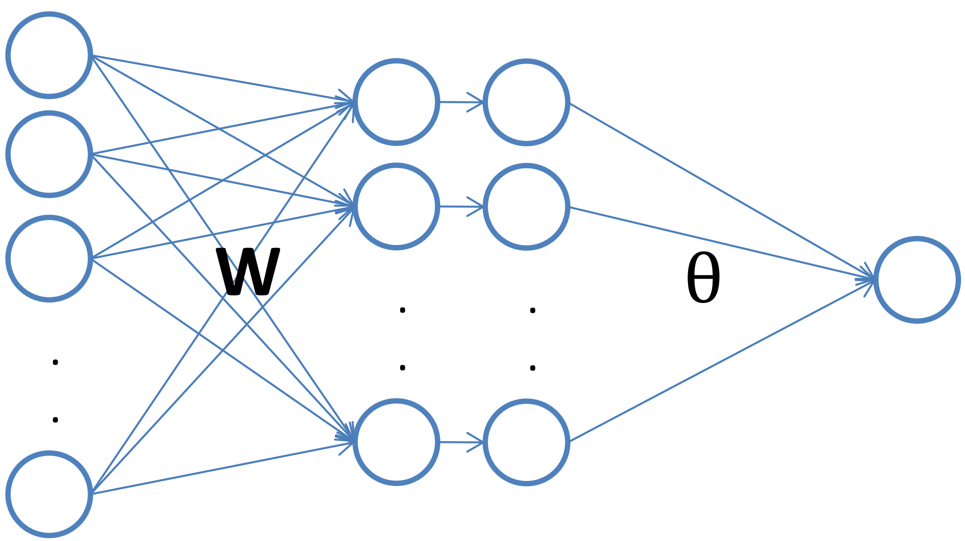
How to Learn the Spatial Filter W?

- **Option C – Bottom-up**: Without looking at the labels y, learn a good spatial filter **W** for the data (unsupervised learning)
- Criterion for a good spatial filter? Independent Component Analysis, Dictionary Learning, PCA



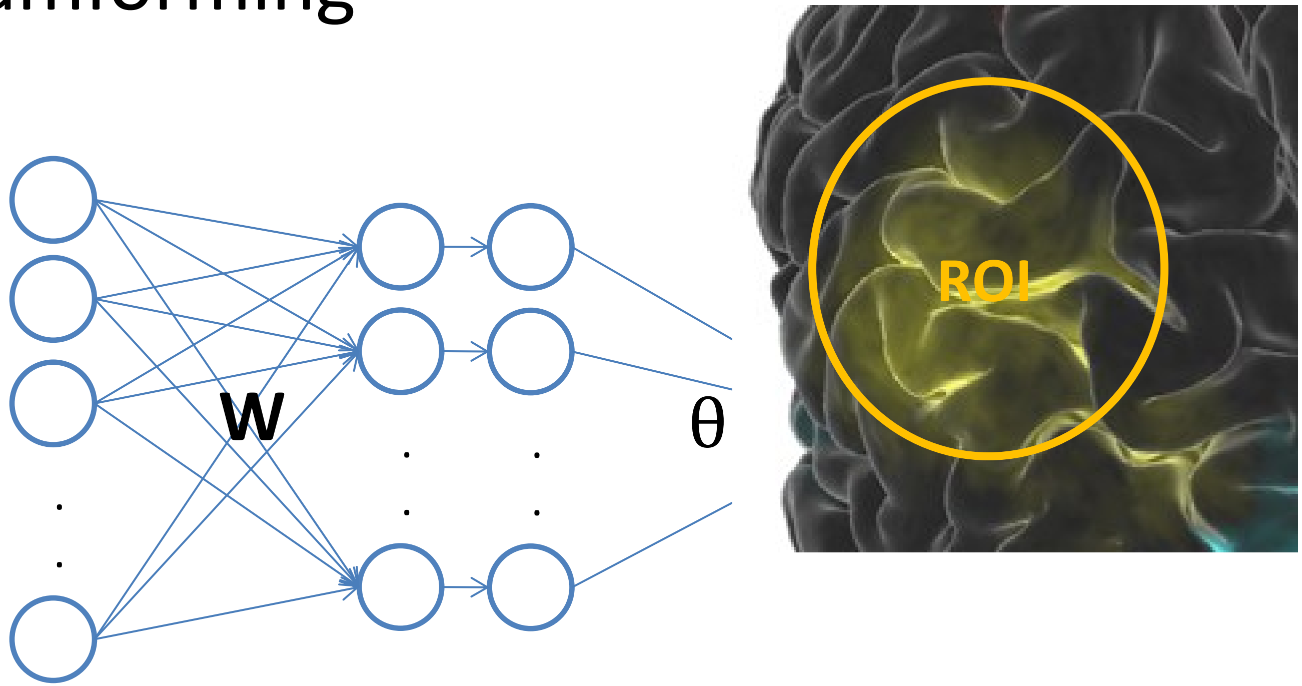
How to Learn the Spatial Filter W?

- **Option D – Both**: Perform a mixture of unsupervised and supervised learning
- Supervised ICA, Unsupervised pre-training + supervised fine-tuning, …



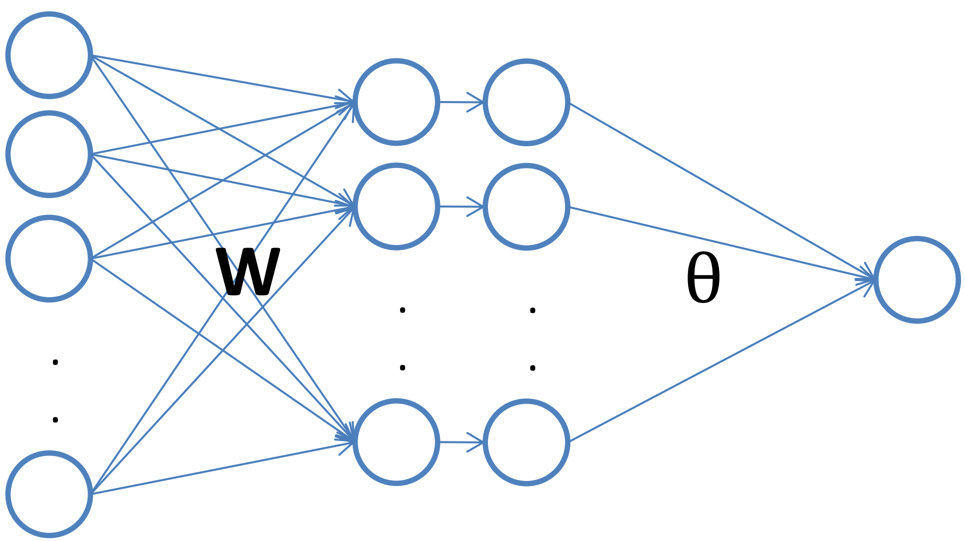
How to Learn the Spatial Filter W?

- **Option E – Using Direct Observations**: Is there a way to observe W directly from data?
- If given an MR scan (or default image), can use e.g., Beamforming



How to Learn the Spatial Filter W?

- **Option F – Using Additional Assumptions**: These can make the problem solvable
- Powerful assumption: the source activation in the time window of interest is *jointly Gaussian-distributed*



Swartz
Center for
Computational
Neuroscience

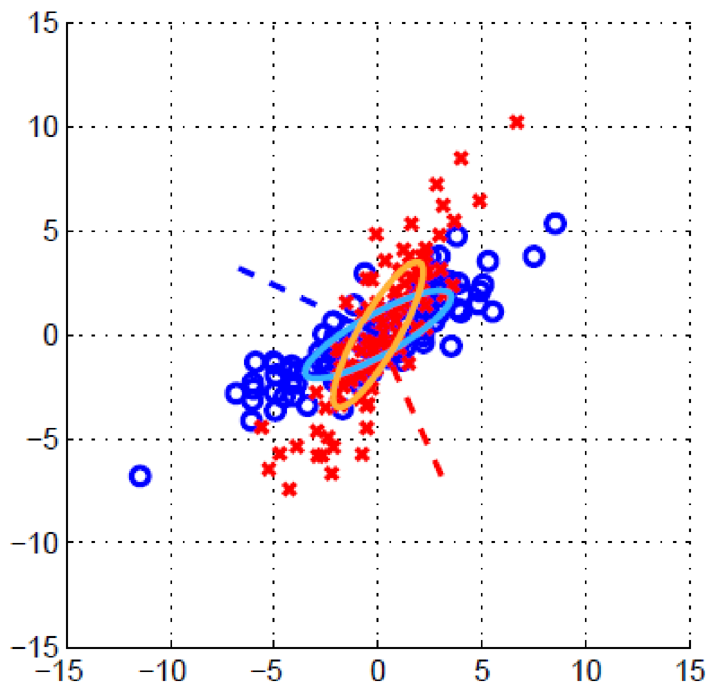
7.3 Common Spatial Patterns

Common Spatial Patterns

- Most popular algorithm in BCI field for learning spatial filters for oscillatory processes
- **Assumptions:**
  - Frequency band and time window are known
  - band-passed signal is jointly Gaussian within the time window
  - Source activity constellation differs between two classes

Common Spatial Patterns

- Different source activities for a left-hand epoch vs. a right-hand epoch (band-passed to 7-30 Hz)
- Signal activation is scatter-plotted for channels C3 and C4:

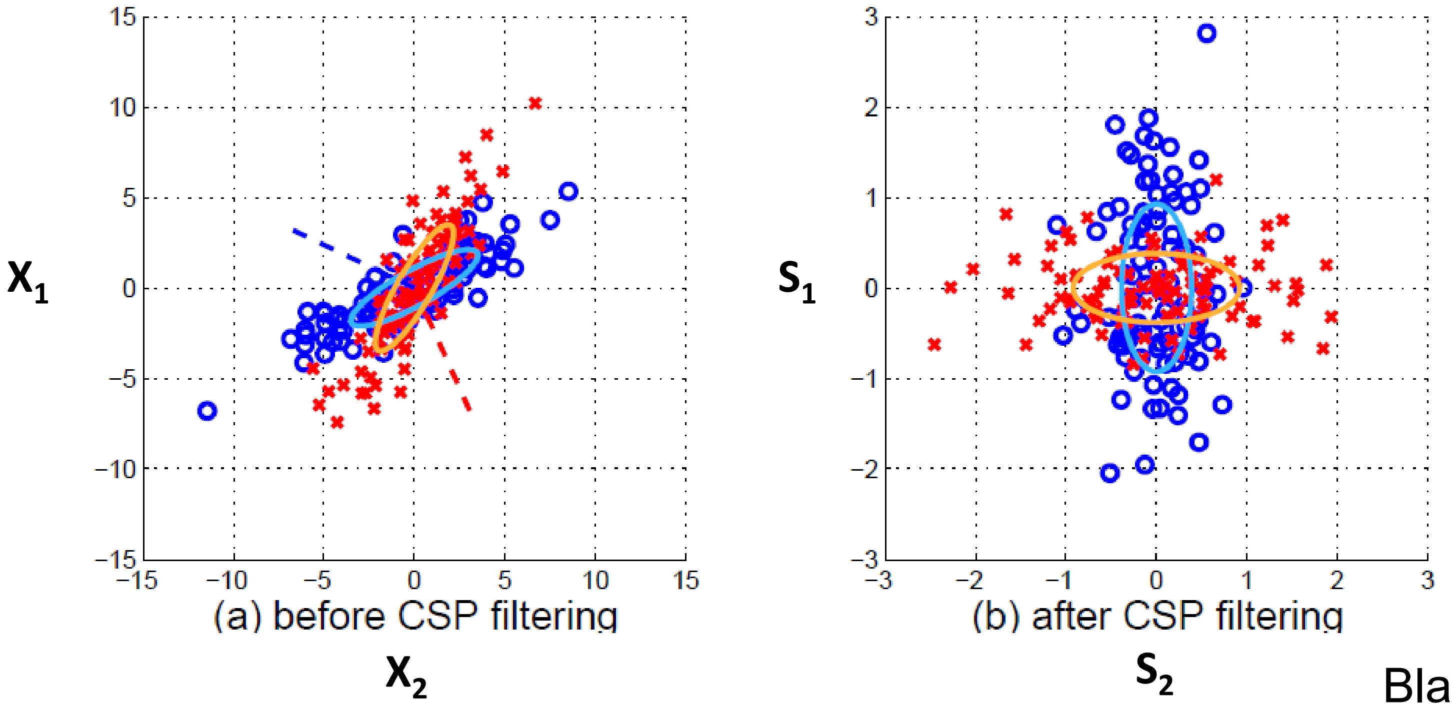


(a) before CSP filtering

Blankertz 2009

Common Spatial Patterns

- Goal: Design a pair of spatial filters (i.e., spatial transforms) such that the filtered signal's variance is maximal for one class while minimal for the other
- And vice versa



Blankertz 2009

Three Ways to Compute It

- **A) Optimization Problem:** Given a set of t trial segments $\boldsymbol{X}_t \in \mathbb{R}^{d \times N}$, per-trial covariance matrices $\boldsymbol{\Sigma}_t = \boldsymbol{X}_t {\boldsymbol{X}_t}^\intercal \in \mathbb{R}^{d \times d}$, and per-class average covariance matrices $\boldsymbol{\Sigma}^{(c)} = \langle \boldsymbol{\Sigma}_t \rangle^c$, optimize the spatial filter $\boldsymbol{w}_c$ for class c as:

$$\boldsymbol{w}_c = \max_{\boldsymbol{w}} \; \boldsymbol{w}^\intercal \boldsymbol{\Sigma}^{(c)} \boldsymbol{w} \;\text{ s.t. }\; \boldsymbol{w}^\intercal \left(\boldsymbol{\Sigma}^{(-1)} + \boldsymbol{\Sigma}^{(+1)} \right) \boldsymbol{w} = 1$$

Three Ways to Compute It

- **A) Optimization Problem:** Given a set of t trial segments $\boldsymbol{X}_t \in \mathbb{R}^{d\times N}$, per-trial covariance matrices $\boldsymbol{\Sigma}_t = \boldsymbol{X}_t {\boldsymbol{X}_t}^\intercal \in \mathbb{R}^{d\times d}$, and per-class average covariance matrices $\boldsymbol{\Sigma}^{(c)} = \langle \boldsymbol{\Sigma}_t \rangle^c$, optimize the spatial filter $\boldsymbol{w}_c$ for class c as:

$$\boldsymbol{w}_c = \max_{\boldsymbol{w}} \ \boldsymbol{w}^\intercal \boldsymbol{\Sigma}^{(c)} \boldsymbol{w} \ \text{ s.t. } \ \boldsymbol{w}^\intercal \left(\boldsymbol{\Sigma}^{(-1)} + \boldsymbol{\Sigma}^{(+1)} \right) \boldsymbol{w} = 1$$

$\boldsymbol{w}^\intercal \boldsymbol{\Sigma} \boldsymbol{w}$ yields the variance in direction $\boldsymbol{w}$

Three Ways to Compute It

- **B) Generalized Eigenvalue Problem:** Given per-class avg. covariance matrices $\boldsymbol{\Sigma_c}$, find the simultaneous diagonalizer $\boldsymbol{V}$ of $\boldsymbol{\Sigma_{-1}}$ and $\boldsymbol{\Sigma_{+1}}$:

$$\boldsymbol{V}^\intercal \boldsymbol{\Sigma}_{-1} \boldsymbol{V} = \boldsymbol{D}_{-1},$$
$$\boldsymbol{V}^\intercal \boldsymbol{\Sigma}_{+1} \boldsymbol{V} = \boldsymbol{D}_{+1},$$

  for diag. $\boldsymbol{D}_{-1}$ and $\boldsymbol{D}_{+1}$ such that $\boldsymbol{D_{-1}} + \boldsymbol{D_{+1}} = \boldsymbol{I}$.
- This yields a generalized eigenvalue problem of the form

$$\boldsymbol{V}^\intercal \boldsymbol{\Sigma}_{-1} \boldsymbol{V} = \boldsymbol{D} \ \wedge \ \boldsymbol{V}^\intercal (\boldsymbol{\Sigma}_{-1} + \boldsymbol{\Sigma}_{+1}) \boldsymbol{V} = \boldsymbol{I}$$

Three Ways to Compute It

- Given the generalized eigenvalue problem of the form

$$\boldsymbol{V}^{\top}\boldsymbol{\Sigma}_{-1}\boldsymbol{V} = \boldsymbol{D} \ \wedge \ \boldsymbol{V}^{\top}(\boldsymbol{\Sigma}_{-1} + \boldsymbol{\Sigma}_{+1})\boldsymbol{V} = \boldsymbol{I}$$

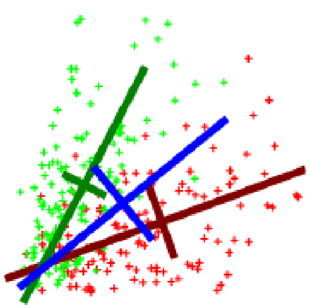
- The k smallest and largest eigenvalues in $\boldsymbol{D}$ correspond to k leftmost/rightmost columns in $\boldsymbol{V}$ (spatial filters) that yield smallest (largest) variance in class -1 and simultaneously largest (smallest) variance in class +1
- Very easy in MATLAB:

```
>> [V,D] = eig(cov1,cov1+cov2)
```

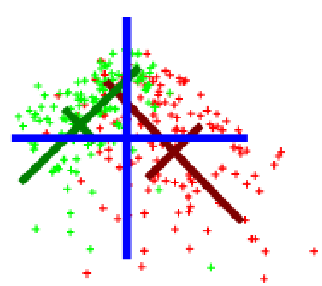
Three Ways to Compute It

- **C) Geometric Approach:** A more intuitive approach is a three-step procedure:
  1. Determine a *whitening* transform ***U*** for the average of both covariance matrices (blue) using PCA
  2. Apply it to one of the matrices and calculate its principal components ***P*** (green)
  3. The spatial filter operation ***W*** is to first whiten by ***U*** and then transform by $\boldsymbol{P}^{-1}$, i.e. $\boldsymbol{W} = \boldsymbol{P}^{-1}\boldsymbol{U}$ so then $\boldsymbol{S} = \boldsymbol{W}\boldsymbol{X}$

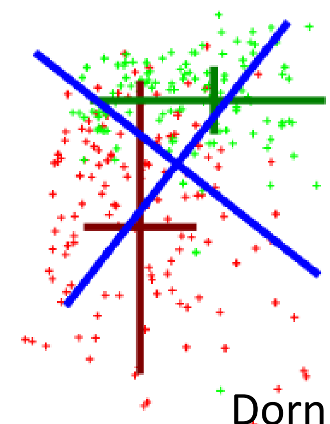
1.



2.



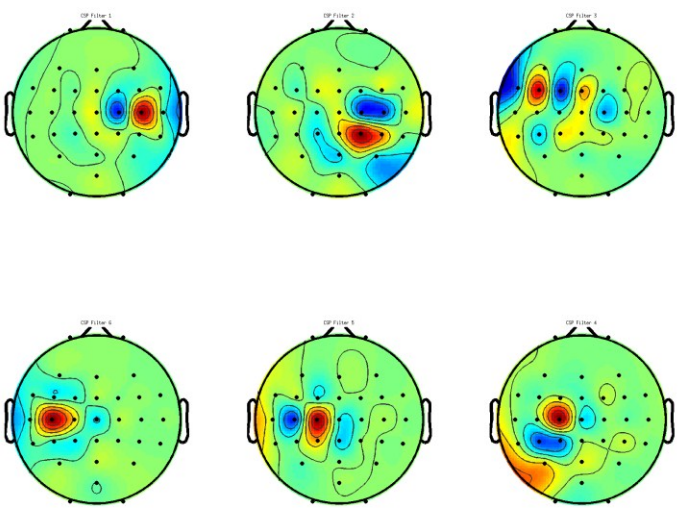
3.



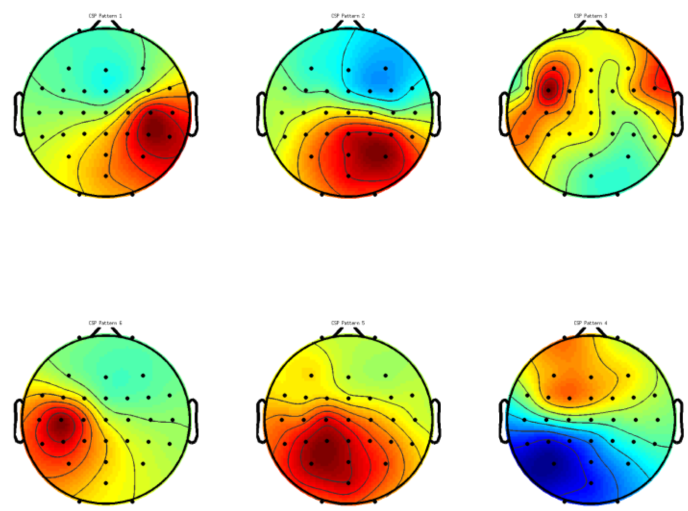
Dornhege, 2004

Resulting Spatial Filters

- Produces well-adapted filters (left) and occasionally roughly dipolar filter inverses (right)
- Note that typically only filters for the k top and k bottom eigenvalues are retained



W



W^{-1}

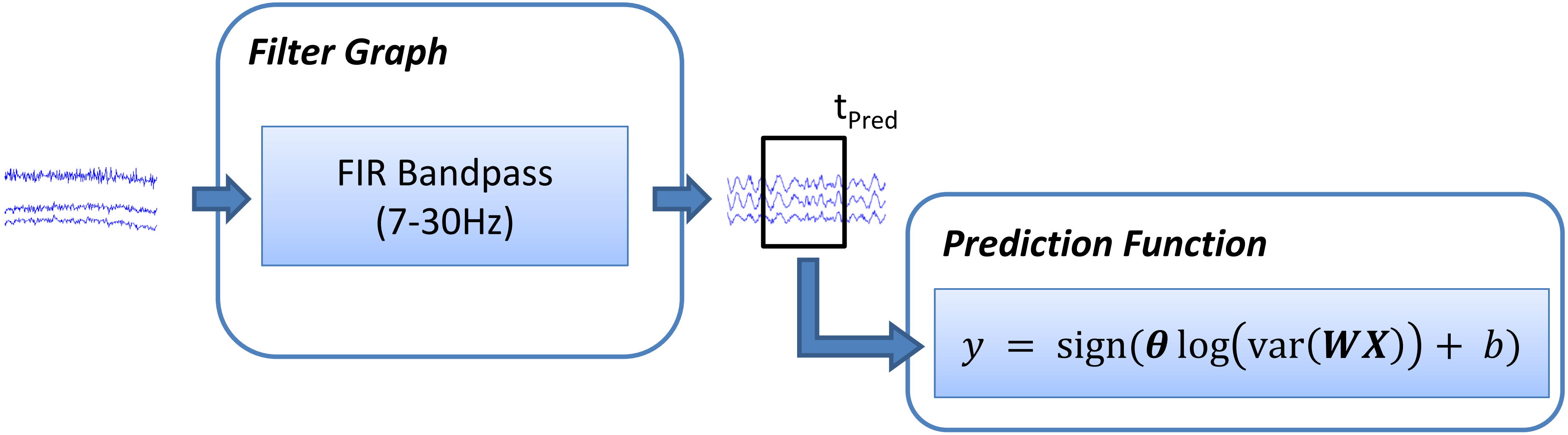
CSP Prediction Function

- The CSP Prediction function amounts to:
  - Spatial filtering
  - Log-variance calculation
  - Application of a linear (or non-linear) classifier

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

Putting it all Together

- A CSP-based BCI typically operates on a band-pass filtered signal
- Choice of the frequency band is not trivial
- The online window length does not need to correspond to the training window length



Swartz
Center for
Computational
Neuroscience

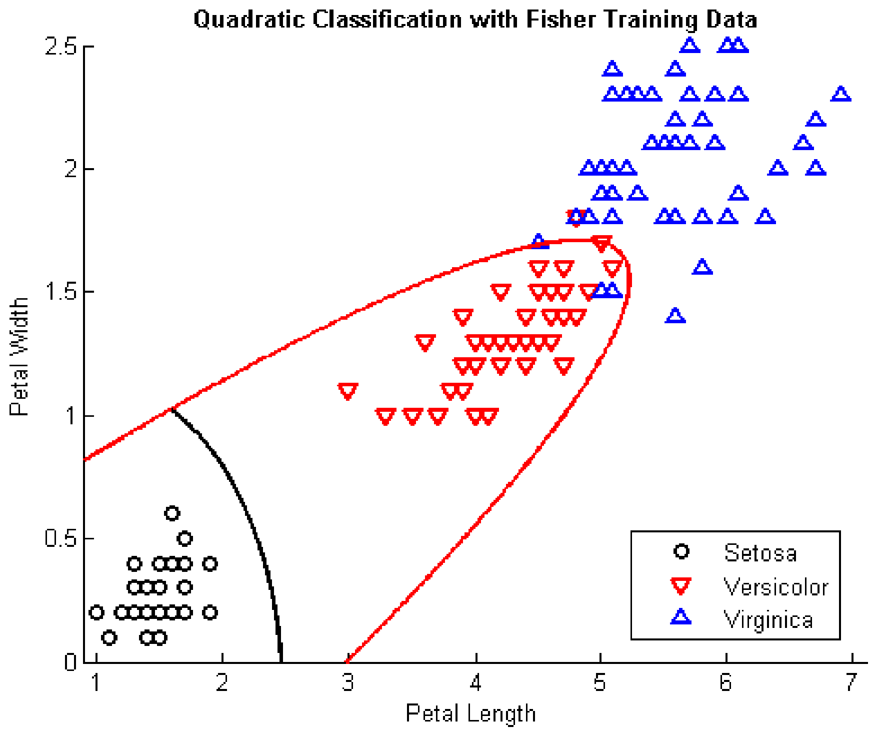
7.4 Alternatives and Extensions

Choosing the Classifier

- Feature space is low-dimensional (4-6) and distributions are well-behaved
- Simple linear classifiers perform well, LDA is hard to beat in practice (strong assumptions)
- Some groups prefer Quadratic Discriminant Analysis (QDA) or other classifiers

Alternatives To LDA

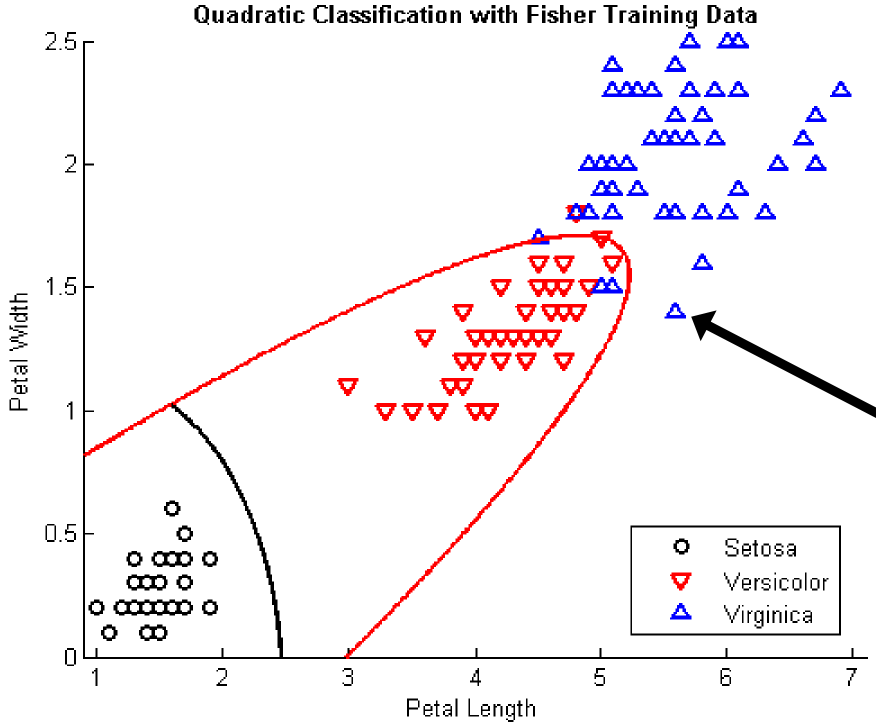
- Omitting the assumption of *condition-independent noise* yields Quadratic Discriminant Analysis (QDA)



The Mathworks

Alternatives To LDA

- Omitting the assumption of *condition-independent noise* yields Quadratic Discriminant Analysis (QDA)
- Surprisingly(?), QDA very rarely performs better than LDA

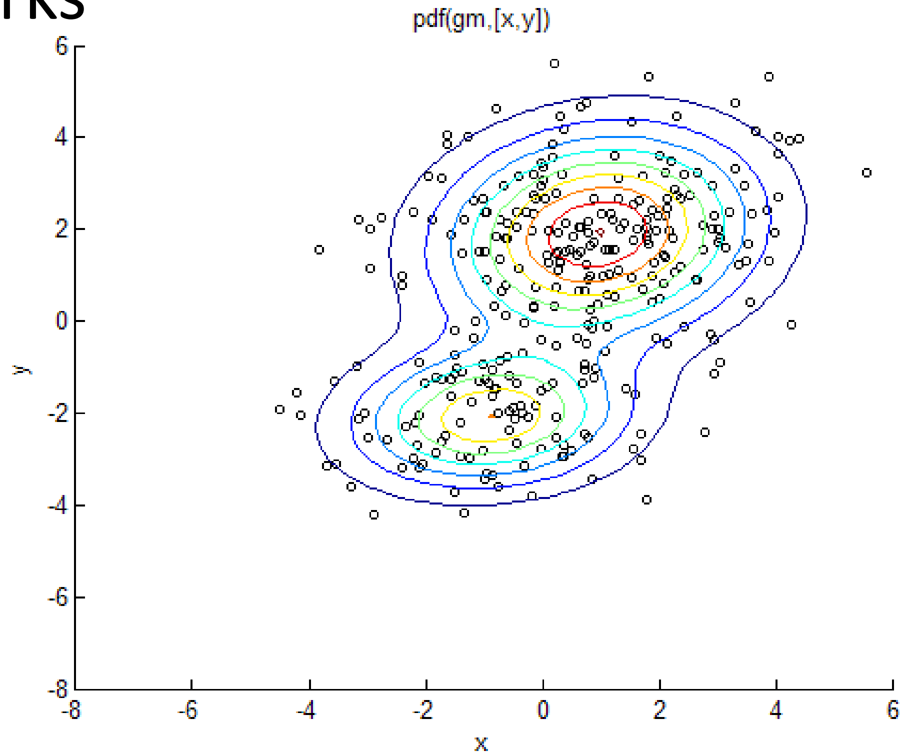


One outlier can cause this

The Mathworks

Alternatives To LDA

- Fitting multiple Gaussians for each condition instead of one yields Gaussian Mixture Models (GMMs)
- GMMs serve as a good "low anchor" in BCI benchmarks



The Mathworks

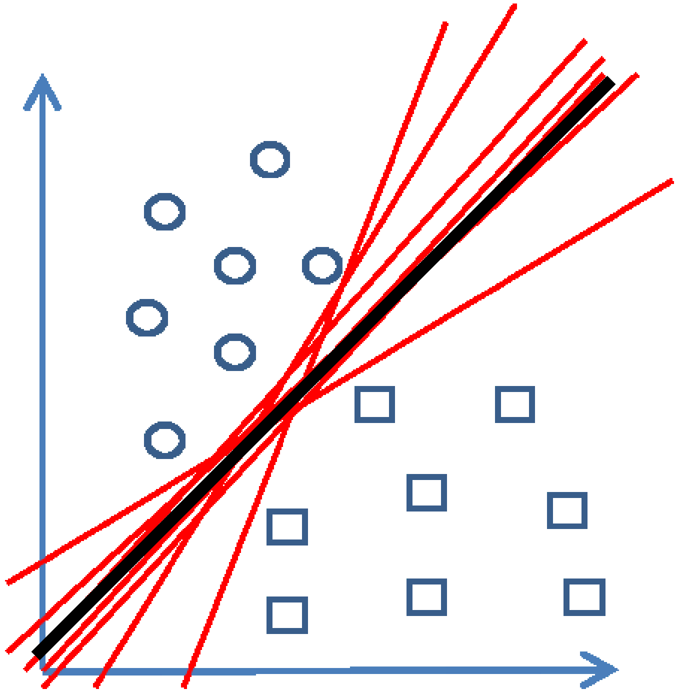
Alternatives To LDA

- Fitting multiple Gaussians for each condition instead of one yields Gaussian Mixture Models (GMMs)
- GMMs serve as a good “low anchor” in benchmarks
- Note that there is *no efficient procedure* to calculate the *globally optimal* GMM fit
- The number of Gaussians is usually not known in advance (unless given by sub-conditions), but can be estimated using Bayesian methods (e.g., VDPGM*) or found via parameter search

*: VDPGM = Variational Dirichlet Process Gaussian Mixture Models

Alternatives To LDA

- **Important Type of Classifiers:** Discriminative models (as opposed to Generative)
- Discussed in the next lecture



Alternatives and Extensions

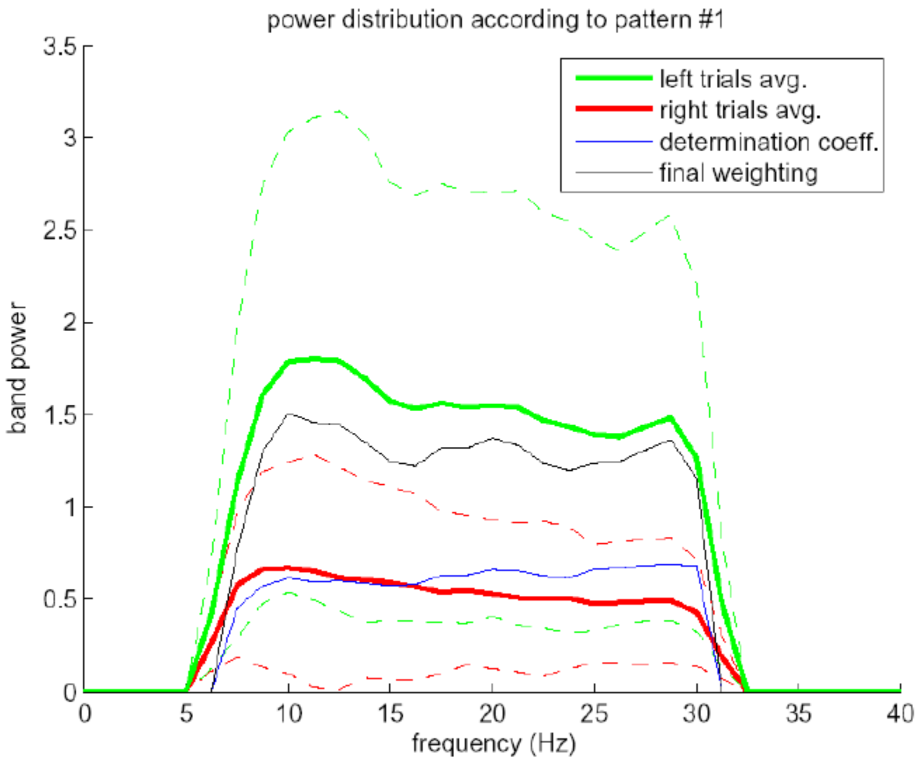
- CSP is the most popular spatial filtering method in the BCI field for oscillations
- There exist >20 extensions addressing various limitations (frequency bands, time window, …)
- The most successful variants so far:
  - Spectrally Weighted CSP (adaptive spectral bands)
  - Filter-Bank CSP (multiple time/frequency windows and *feature combination*)
  - Regularized CSPs (if too few training trials)

Spectrally Weighted CSP

- One of the best algorithms for learning the correct frequency bands (others: CSSP, CSSSP)
- Iterative algorithm that alternates between optimizing the spatial and spectral filters (block coordinate descent)
  - Spatial filters are optimized using CSP
  - Spectral filters are optimized using Person's correlation coefficient

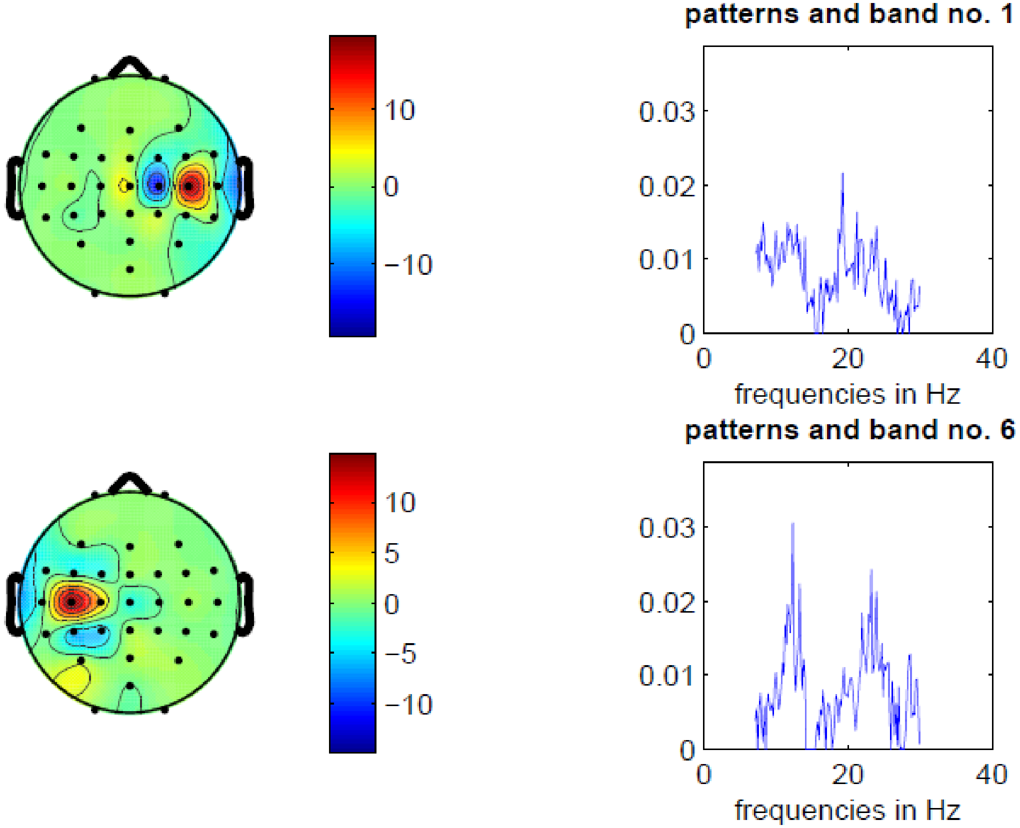
Spectrally Weighted CSP

- Updating a spectral filter given spatially filtered data:



Resulting Spatio-Spectral Filters

- Adaptive filters for left vs. right hand movement imagination:



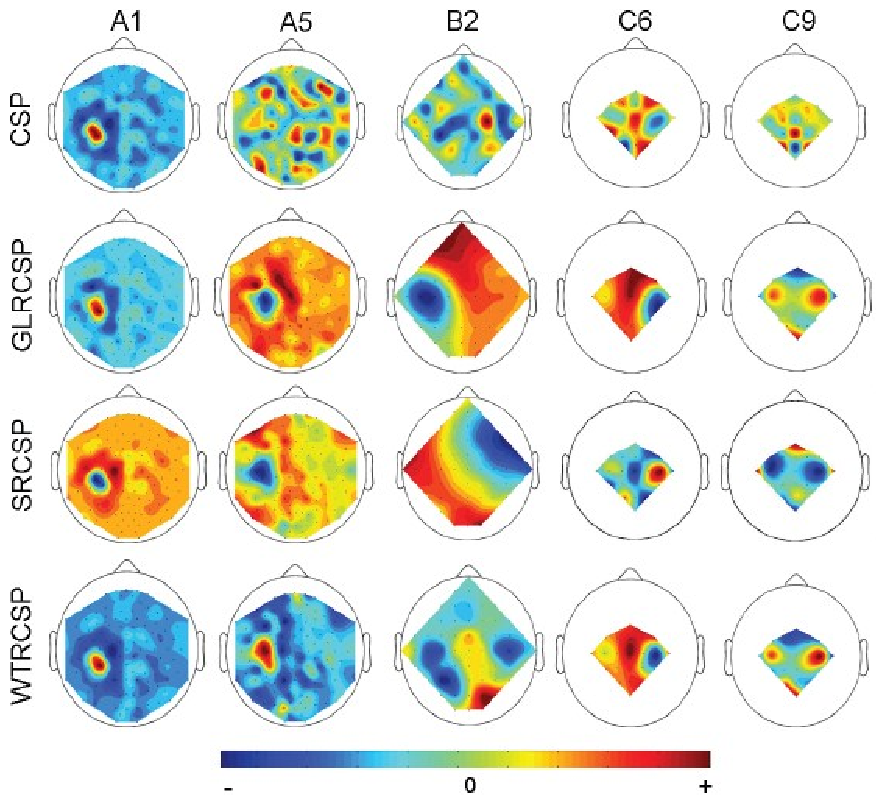
Spec-CSP Prediction Function

- For simplicity here implemented without any signal processing stage, using a temporal filter matrix (not very efficient):
  - **S**: spatial filters
  - **T**: temporal filters
  - $\boldsymbol{\theta}$, b: linear classifier
- Can be used for any other CSP-like approach that requires a temporal filter

$$y = \operatorname{sign}(\boldsymbol{\theta} \log\left(\operatorname{var}\left((\boldsymbol{SX})\boldsymbol{T}\right)\right) + b)$$

Regularized CSP Variants

- Add a regularization term and parameter that needs to be searched via grid search
- Compared methods:
  - Basic CSP
  - Generic Learning CSP
  - Spatially Regularized CSP
  - Weighted Tikhonov-Regularized CSP
- 5 pathological data sets (from BCI competitions)



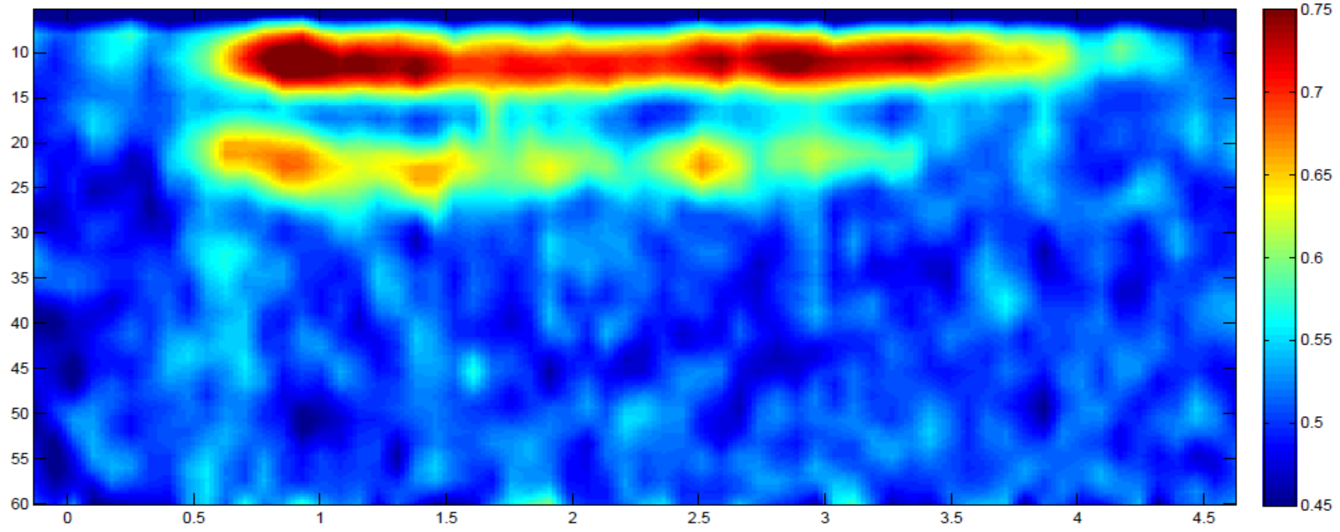
Lotte et al., 2010

Multi-Class Extensions

- Most CSP variants are inherently defined for two classes
- Not a problem – can solve CSP for pairs of classes, train pairwise classifiers, and determine most likely class by voting
- Possibilities: one-versus-rest, one-versus-one
- **Note:** classifiers should preferably produce probabilistic outputs so that voting can be done as probabilistically evidence gathering
- A useful prob. classifier is *logistic regression*

Time Window Estimation

- The time window is a free parameter that depends on the task of interest
- Can be searched as a 2d parameter space (slow!)
- Can be chosen via heuristics (threshold the correlation coefficients – or use them as weighting)



L 7 Questions?