Preprocessing and Classification of ERD/ERS Signals

Florian Eichin
Advisor: Andreas Meinel

July 17, 2018

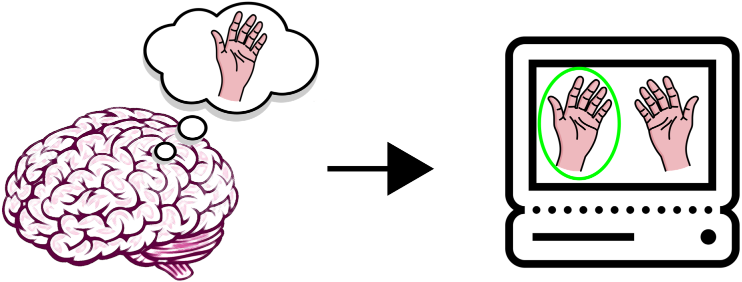
Freiburg University

Contents of this Talk

1. How do we measure brain activity?
2. Oscillatory Analysis
3. Noise, Dimension, Localization
4. Spatial Filters
   - Laplace Filters
   - Common Spatial Patterns (CSP)
5. Classifying ERD/ERS

Goal/Motivation for this talk

Via brain activity: Classify (imagined) left/right hand movement

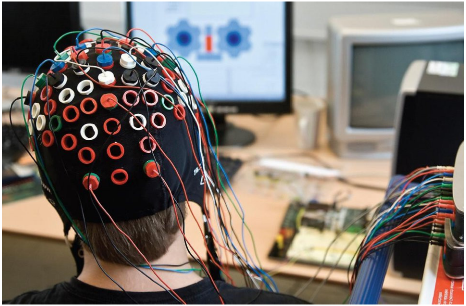


How can we achieve this?

How do we measure brain activity?

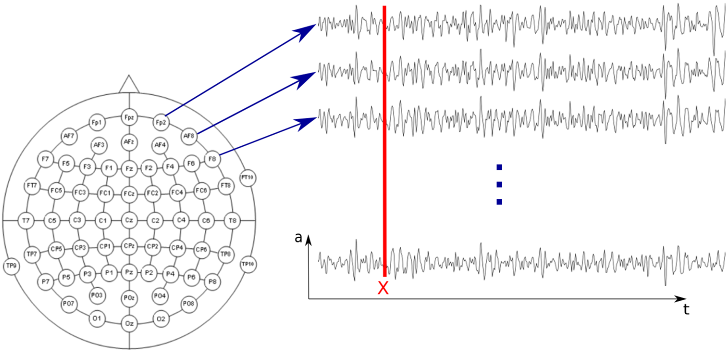
Which technology do we use?

- Many ways to measure brain activity:
  - fMRI, fNIR
  - EEG, MEG, ECoG
- Invasive or non-invasive?
- Trade-off: time and spatial resolution, costs, mobility
- **In this talk: electromagnetic activity via EEG**



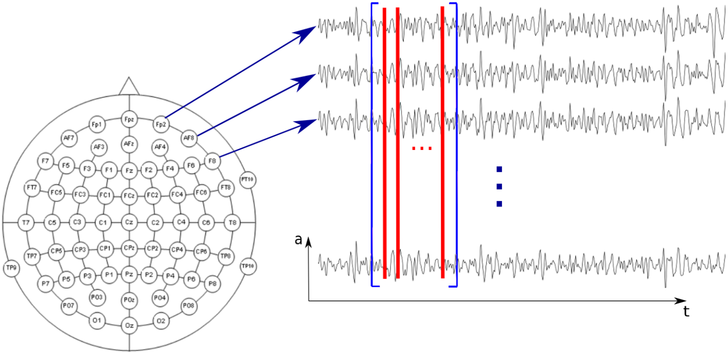
[Wolpaw, 2011]

Representation of EEG-Signals



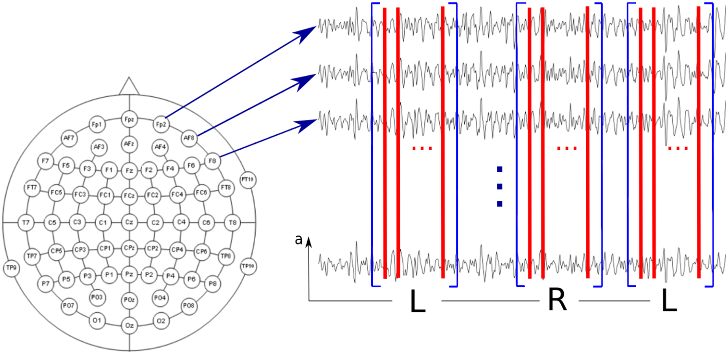
- Let $t, d, C \in \mathbb{N}$
- A **sample** is a vector $\mathbf{x} \in \mathbb{R}^C$
-
-

Representation of EEG-Signals



- Let $t, d, C \in \mathbb{N}$
- A **sample** is a vector $\mathbf{x} \in \mathbb{R}^C$
- $X = (\mathbf{x}^{(t)}\ \mathbf{x}^{(t+1)}\ \ldots\ \mathbf{x}^{(t+d)}) \in \mathbb{R}^{Cxd}$ is called an **epoch**
-

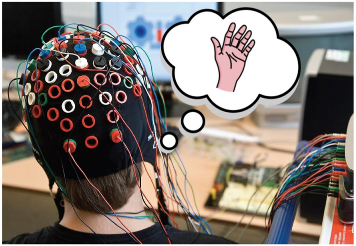
Representation of EEG-Signals

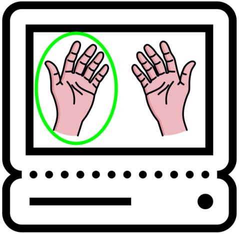


- Let $t, d, C \in \mathbb{N}$
- A **sample** is a vector $\mathbf{x} \in \mathbb{R}^C$
- $X = (\mathbf{x}^{(t)}\ \mathbf{x}^{(t+1)}\ \dots\ \mathbf{x}^{(t+d)}) \in \mathbb{R}^{Cxd}$ is called an **epoch**
- We will use labeled epochs for training our classifier

Next step to our goal

Via brain activity: Classify (imagined) left/right hand movement





What features of measured brain activity are useful?

Oscillatory Analysis: Some Background

Pyramidal Cells - Generators of EEG

- Largest contributor to electromagnetic activity
- Aligned orthogonally to the cortex surface
- Electromagnetic fields add up whenever we have co-aligned, co-activated activity

→ **Only large scale, synchronous activity picked up by EEG**

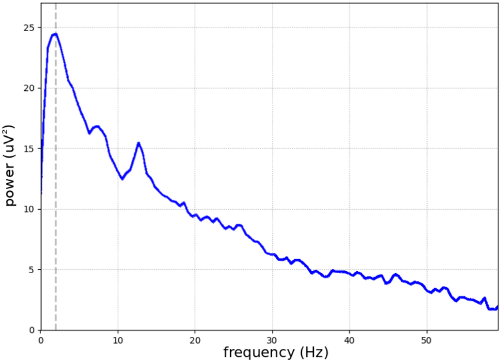
pyramidal_alignedpm.png

[Aarhuus University, 2004]

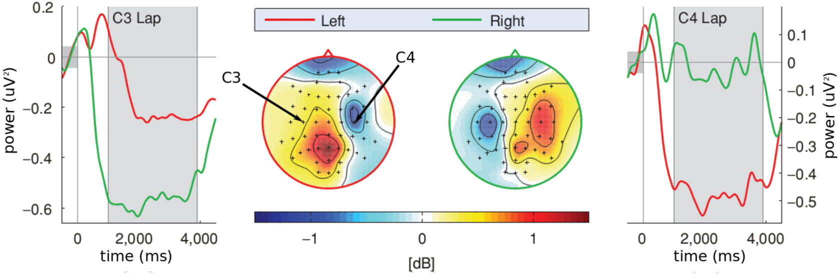
Idle Oscillations - What we assume

- Inactive populations of neurons enter an **idle state**
- They fire synchronously at characteristic frequencies

  e.g. α- / μ-rhythms: 8-15 Hz, found in visual / motor cortex
- Can be observed on frequency spectrum:



Event Related (De)Synchronizations



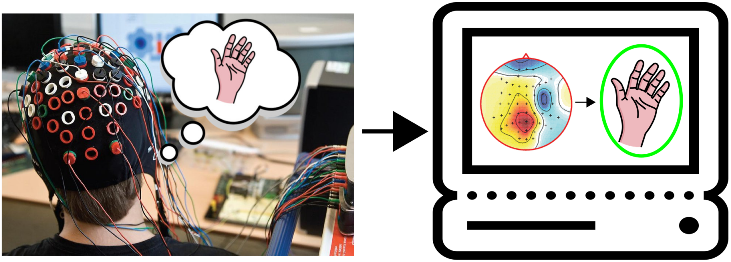
[Blankertz, 2008]

- increase of power = Event Related Synchronization (**ERS**)
- decrease of power = Event Related Desynchronization (**ERD**)
- Parts of the brain are linked to certain tasks

→ Local ERD/ERS will help discriminating left and right hand

We're still looking at Problems

Via brain activity: classify (imagined) left/right hand movement



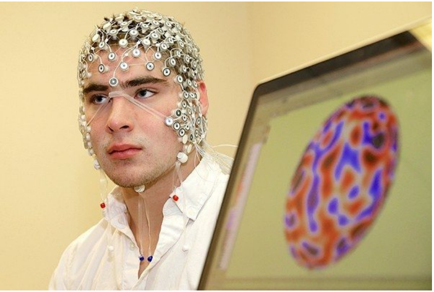
Why don't we just look for local ERD/ERS?

Noise, Dimension and Localization: What are the challenges?

Noise and Dimension

- Raw EEG-data has low signal-to-noise ratio
- Also high dimensionality (up to 128 channels)

→ **We need a lot of training data**



[University of Nebraska, 2013]

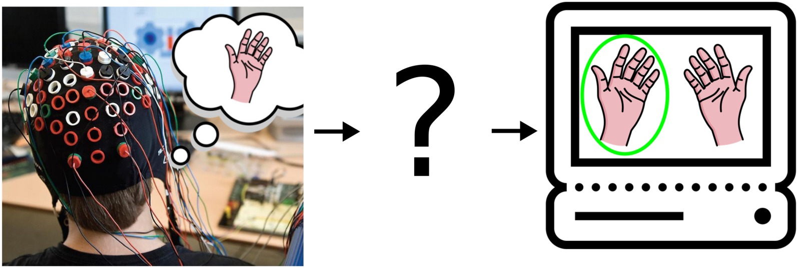
Localization of ERD/ERS activity

- Brain processes have high subject-to-subject variation
- Even with the same subject we will have high session-to-session variation

$\rightarrow$ **New session/subject: Localize relevant ERD/ERS again**

We need Preprocessing

Via brain activity: Classify (imagined) left/right hand movement



How can we improve signal-to-noise ratio, decrease dimensionality and deal with localization?

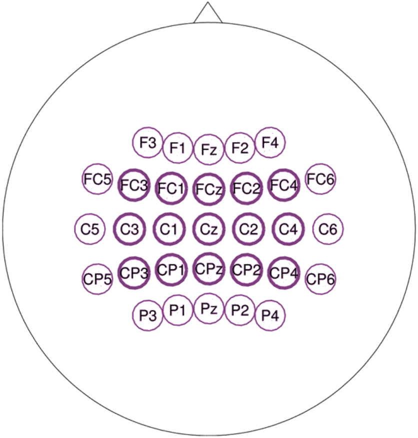
Simple solution: Laplace Filters

What is a Spatial Filter?

- C the number of channels
- A **spatial filter** is a vector $\mathbf{w} \in \mathbb{R}^C$
- For a sample $\mathbf{x} \in \mathbb{R}^C$ the filtered sample is

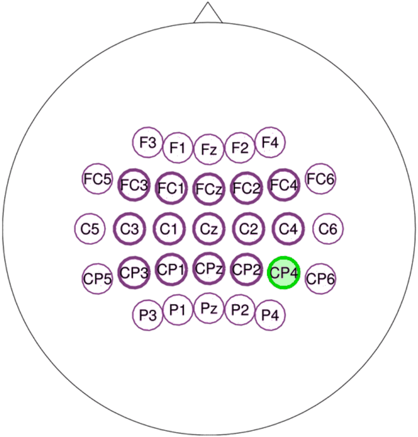
$$\tilde{x} = \mathbf{w}^T \cdot \mathbf{x} \in \mathbb{R} \tag{1}$$

Laplace Filters



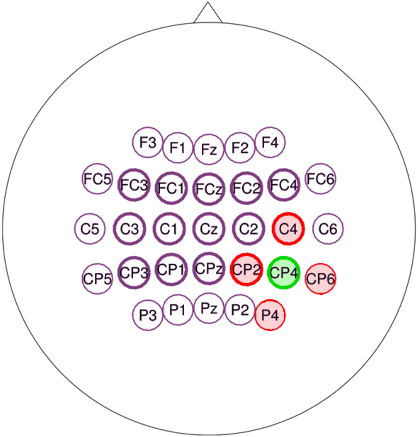
$$\mathbf{x} = (x_{F3}, ..., x_{P4})^T$$

Laplace Filters



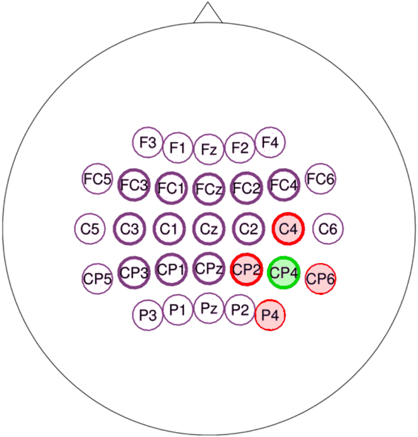
$\mathbf{x} = (x_{F3}, ..., x_{P4})^T$

Laplace Filters



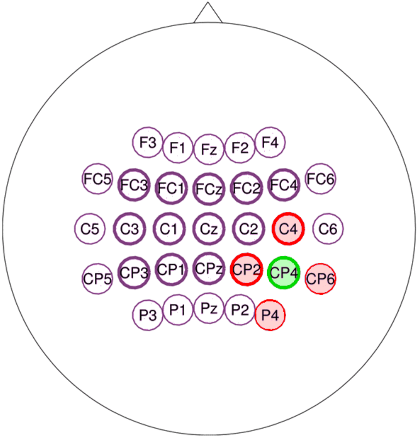
$\mathbf{x} = (x_{F3}, \ldots, x_{P4})^T$

Laplace Filters



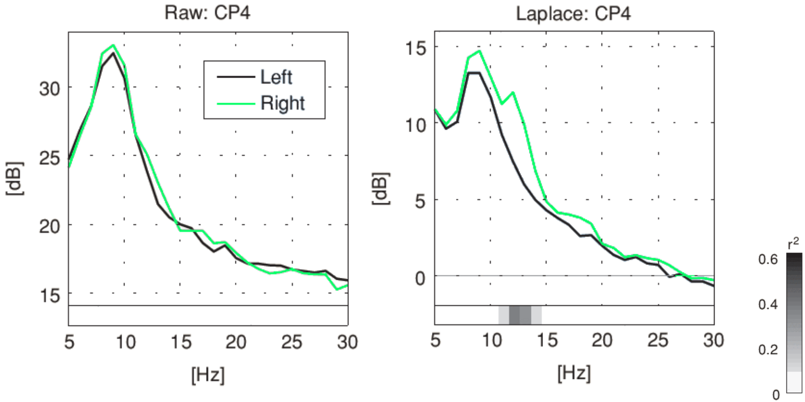
$$\mathbf{x} = (x_{F3}, \ldots, x_{P4})^T$$
$$\mathbf{w} = (0, \ldots, 0, -\frac{1}{4}, -\frac{1}{4}, 1, -\frac{1}{4}, -\frac{1}{4})^T$$

Laplace Filters



$$\mathbf{x} = (x_{F3}, ..., x_{P4})^T$$
$$\mathbf{w} = (0, ..., 0, -\frac{1}{4}, -\frac{1}{4}, 1, -\frac{1}{4}, -\frac{1}{4})^T$$
$$\tilde{x} = \mathbf{w}^T \cdot \mathbf{x} = -\frac{1}{4} \cdot \left(x_{C4} + x_{CP2} + x_{CP6} + x_{P4}\right) + x_{CP4}$$

What can be achieved?



[Blankertz, 2008]

- Slight peak in μ-range at around 12 Hz

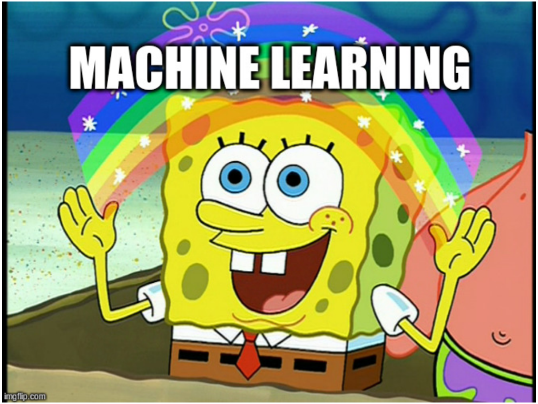
$\rightarrow$ r^2-score indicates: already useful discriminability

Are Laplace Filters what we searched for?

Solution	SNR	Dimensionality	Localization
Laplace Filters	Somewhat	Yes	By hand

How can we (further) improve signal/noise ratio, decrease dimensionality and deal with localization?

Are Laplace Filters what we searched for?

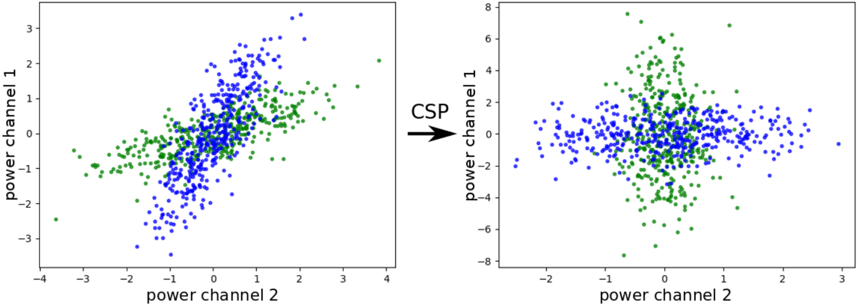


How can we (further) improve signal/noise ratio, decrease dimensionality and deal with localization?

A data-driven approach:
Common Spatial Patterns (CSP)

Common Spatial Patterns (CSP)

- Learn filters from sample epochs $X_i \in \mathbb{R}^{CxD}$ for class L and R
- Variance in EEG channels estimates average power of signal
- Idea: **Contrast variance** between the two classes



CSP Step I - Covariance Estimation

- Let $X_1, ..., X_k \in \mathbb{R}^{CxD}$ be the sample epochs for class h
- Then we estimate the **covariance matrix** by

$$\Sigma_h = \frac{1}{k} \sum_{i=1}^{k} X_i X_i^T \tag{2}$$

CSP Step II - Optimization Problem

- Let $\Sigma_L, \Sigma_R \in \mathbb{R}^{CxC}$ be the covariance matrix for class L / R
- Subsequently find orthogonal vectors $\mathbf{w}_i$ that satisfy

$$\mathbf{w}_i = \underset{\mathbf{w} \in \mathbb{R}^C}{argmax} \frac{\mathbf{w}^T \Sigma_L \mathbf{w}}{\mathbf{w}^T \Sigma_R \mathbf{w}} \tag{3}$$

CSP Step II - Optimization Problem

- Let $\Sigma_L, \Sigma_R \in \mathbb{R}^{CxC}$ be the covariance matrix for class L / R
- Subsequently find orthogonal vectors $\mathbf{w}_i$ that satisfy

$$\mathbf{w}_i = \underset{\mathbf{w} \in \mathbb{R}^C}{argmax} \frac{\mathbf{w}^T \Sigma_L \mathbf{w}}{\mathbf{w}^T \Sigma_R \mathbf{w}} \tag{3}$$

- $W = (\mathbf{w}_1 \ ... \ \mathbf{w}_N)^T$ projects data to a space, where the first coordinate has the highest (lowest) variance for class L (R)

$\rightarrow$ in application: only keep first and last couple of vectors

Analytical Solution for the Optimization Problem

This optimization problem can be solved with Lagrange Multipliers:

Analytical Solution for the Optimization Problem

This optimization problem can be solved with Lagrange Multipliers:

$\rightarrow$ **maximize** $\mathbf{w}^T\Sigma_L\mathbf{w}$ **w.r.t.** $\mathbf{w}^T\Sigma_R\mathbf{w} = c$

Analytical Solution for the Optimization Problem

This optimization problem can be solved with Lagrange Multipliers:

$\rightarrow$ **maximize** $\mathbf{w}^T \Sigma_L \mathbf{w}$ **w.r.t.** $\mathbf{w}^T \Sigma_R \mathbf{w} = c$

$$L(\lambda, \mathbf{w}) = \mathbf{w}^T \Sigma_L \mathbf{w} - \lambda(\mathbf{w}^T \Sigma_R \mathbf{w} - c)$$

Analytical Solution for the Optimization Problem

This optimization problem can be solved with Lagrange Multipliers:

$\rightarrow$ **maximize** $\mathbf{w}^T \Sigma_L \mathbf{w}$ **w.r.t.** $\mathbf{w}^T \Sigma_R \mathbf{w} = c$

$L(\lambda, \mathbf{w}) = \mathbf{w}^T \Sigma_L \mathbf{w} - \lambda(\mathbf{w}^T \Sigma_R \mathbf{w} - c)$

$\frac{\partial}{\partial \mathbf{w}} L(\lambda, \mathbf{w}) = 2\Sigma_L \mathbf{w} - \lambda 2 \Sigma_R \mathbf{w}$

Analytical Solution for the Optimization Problem

This optimization problem can be solved with Lagrange Multipliers:

$\rightarrow$ **maximize** $\mathbf{w}^T\Sigma_L\mathbf{w}$ **w.r.t.** $\mathbf{w}^T\Sigma_R\mathbf{w} = c$

$L(\lambda, \mathbf{w}) = \mathbf{w}^T\Sigma_L\mathbf{w} - \lambda(\mathbf{w}^T\Sigma_R\mathbf{w} - c)$

$\frac{\partial}{\partial \mathbf{w}} L(\lambda, \mathbf{w}) = 2\Sigma_L\mathbf{w} - \lambda 2\Sigma_R\mathbf{w} \overset{!}{=} 0$

Analytical Solution for the Optimization Problem

This optimization problem can be solved with Lagrange Multipliers:

$\rightarrow$ **maximize** $\mathbf{w}^T \Sigma_L \mathbf{w}$ **w.r.t.** $\mathbf{w}^T \Sigma_R \mathbf{w} = c$

$$L(\lambda, \mathbf{w}) = \mathbf{w}^T \Sigma_L \mathbf{w} - \lambda(\mathbf{w}^T \Sigma_R \mathbf{w} - c)$$

$$\frac{\partial}{\partial \mathbf{w}} L(\lambda, \mathbf{w}) = 2\Sigma_L \mathbf{w} - \lambda 2 \Sigma_R \mathbf{w} \overset{!}{=} 0$$

$$\Leftrightarrow \Sigma_L \mathbf{w} = \lambda \Sigma_R \mathbf{w}$$

Analytical Solution for the Optimization Problem

This optimization problem can be solved with Lagrange Multipliers:

$\rightarrow$ **maximize** $\mathbf{w}^T \Sigma_L \mathbf{w}$ **w.r.t.** $\mathbf{w}^T \Sigma_R \mathbf{w} = c$

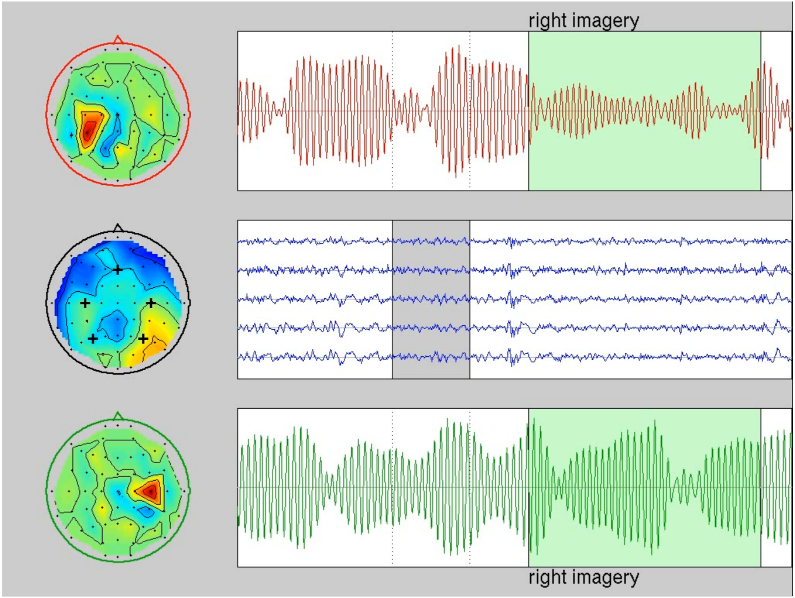
$$L(\lambda, \mathbf{w}) = \mathbf{w}^T \Sigma_L \mathbf{w} - \lambda(\mathbf{w}^T \Sigma_R \mathbf{w} - c)$$

$$\frac{\partial}{\partial \mathbf{w}} L(\lambda, \mathbf{w}) = 2\Sigma_L \mathbf{w} - \lambda 2 \Sigma_R \mathbf{w} \stackrel{!}{=} 0$$

$$\Leftrightarrow \ \Sigma_L \mathbf{w} = \lambda \Sigma_R \mathbf{w}$$

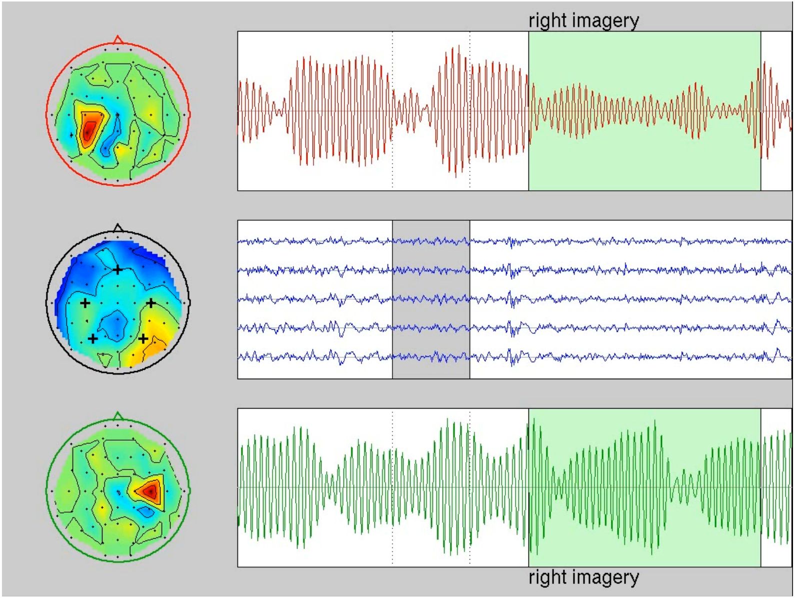
- Which is a **generalized eigenvalue problem**
- Eigenvectors $\mathbf{w}_1, ..., \mathbf{w}_n$ with eigenvalues $\lambda_1 > \lambda_2 > ... > \lambda_n$ yield $W = (\mathbf{w}_1 \ ... \ \mathbf{w}_n)^T$

CSP in Action



[Tangermann, 2018]

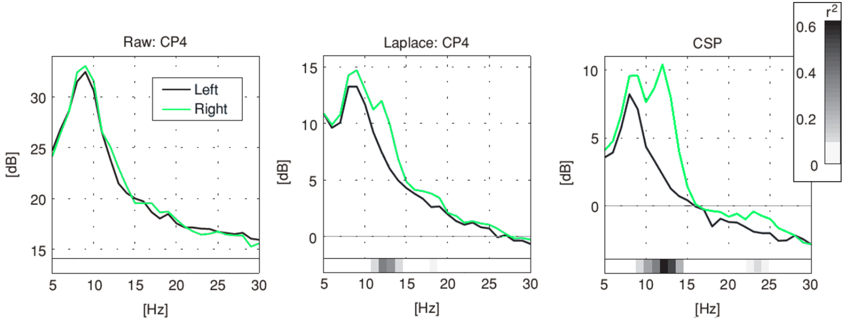
CSP in Action



[Tangermann, 2018]

Evaluation: Even better results

- α- and μ-bands are more prominent after filtering
- r^2-score: Good discriminanility in CSP-channel



[Blankertz, 2008]

Are CSP-filters what we searched for?

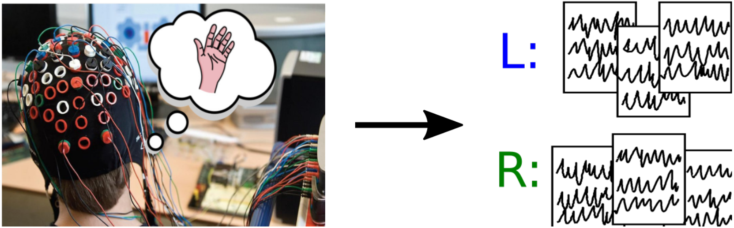
Solution	SNR	Dimensionality	Localization
Laplace Filters	Somewhat	Yes	By hand
CSP	Yes	Yes	Yes

- easy to use and implement
- analytical solution
- low runtime (linear mapping)
- on the other hand:
  - hyperparameters (length, time point of epochs ...)
  - supervised: need for labeled data

Final Step: Classification

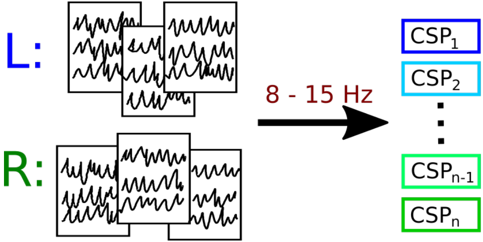
Let's put it together! - Intuition

1. Fetch labeled epochs from participant
2.
3.
4.



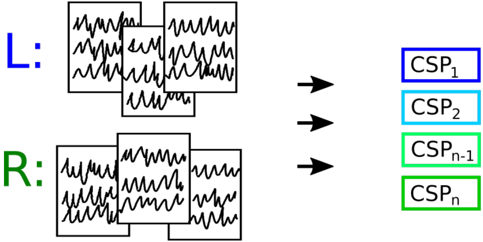
Let's put it together! - Intuition

1. Fetch labeled epochs from participant
2. Compute CSP-filters on bandpassed (e.g. 8-15 Hz) epochs
3.
4.



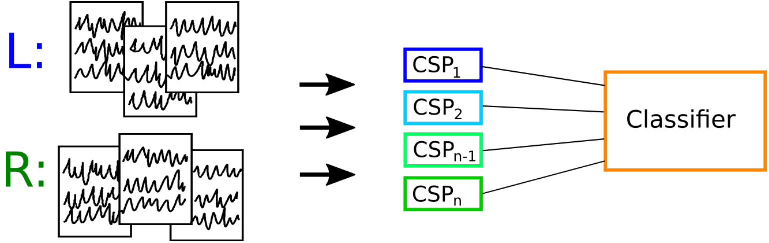
Let's put it together! - Intuition

1. Fetch labeled epochs from participant
2. Compute CSP-filters on bandpassed (e.g. 8-15 Hz) epochs
3. Use the outer CSP-channels (highest contrast in power!)
4.

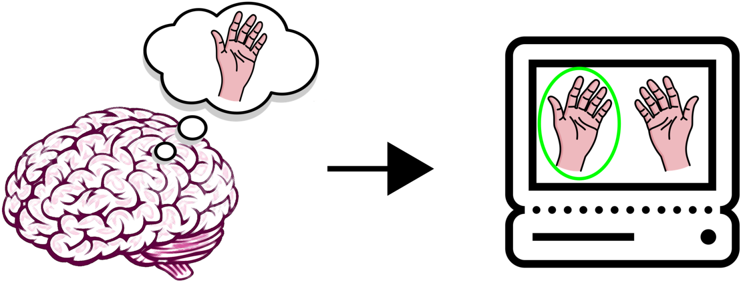


Let's put it together! - Intuition

1. Fetch labeled epochs from participant
2. Compute CSP-filters on bandpassed (e.g. 8-15 Hz) epochs
3. Use the outer CSP-channels (highest contrast in power!)
4. Train classifier on average power of CSP-channels of epoch



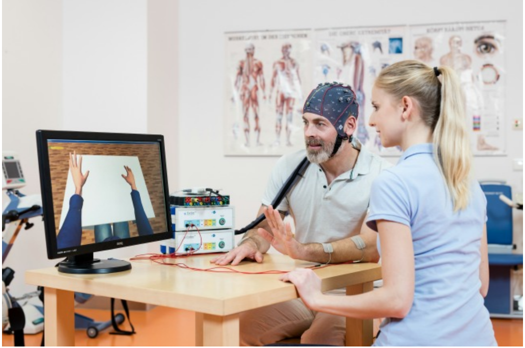
Conclusion - Did we achieve our goal?



- In [1] this approach was used with average accuracy of 90%

$\rightarrow$ **this could be useful for a lot of tasks already**

Conclusion - Did we achieve our goal?



- In [1] this approach was used with average accuracy of 90%

→ **this could be useful for a lot of tasks already**

Conclusion - Did we achieve our goal?

- In [1] this approach was used with average accuracy of 90%

→ **this could be useful for a lot of tasks already**

Sources

[1] Blankertz B, Tomioka R, Lemm S, Kawanabe M, Mueller KR, Optimizing Spatial Filters for Robust EEG Single-Trial Analysis. IEEE Signal Process Mag, 25(1):41-56, 2008.

[2] Wolpaw, Brain-Computer Interfaces: principles and practice. Oxford Univ. Press, 2011.

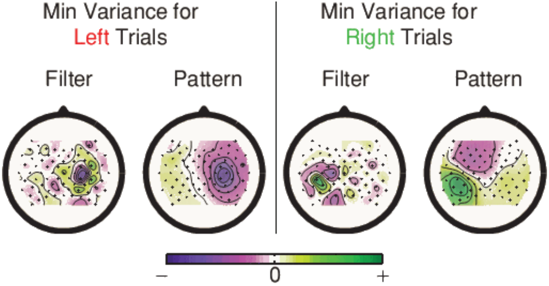
[3] mnetools.github.io/dev/auto_examples/decoding/plot_decoding_csp_eeg.html

[4] Schalk, G McFarland, DJ Hinterberger T, Birbaumer N, Wolpaw JR, BCI2000: A General-Purpose Brain-Computer Interface (BCI) System. IEEE TBME 51(6):1034-1043, 2004

[5] Introduction to Modern Brain-Computer Interface Design - Christian A. Kothe Swartz Center for Computational Neuroscience, University of California San Diego

Filters and Patterns

W consists of filters, the rows of $A = W^{-1}$ are called patterns.



Relation to Principal Component Analysis

CSP is a supervised generalization of PCA for two classes.
If class **R** is is uncorrelated, CSP yields the same filter matrix as PCA for class **L**.

Proof.
Let Σ_L and Σ_R be the covariance matrices for class L and R.
Then $\Sigma_R = I$ and we have

$$w_i = \underset{w \in \mathbb{R}^N}{argmax} \frac{w^T \Sigma_L w}{w^T \Sigma_R w} = \underset{w \in \mathbb{R}^n}{argmax} \frac{w^T \Sigma_L w}{w^T w} \tag{4}$$

which is the optimization criterion for PCA. $\square$

Idle Oscillations - Video

[Backyard Brains, 2014]

Idle Oscillations - Video

[Backyard Brains, 2014]

So what could this classifier look like?

We train and use the following classifier:

$$f(X, \mathbf{w}_1, ..., \mathbf{w}_J, \beta_0, ..., \beta_J) = \sum_{j=1}^{J} \beta_j \log(\mathbf{w}_j^T X X^T \mathbf{w}_j) + \beta_0 \qquad (5)$$

-
-
-
-
-

So what could this classifier look like?

We train and use the following classifier:

$$f(X, \mathbf{w}_1, ..., \mathbf{w}_J, \beta_0, ..., \beta_J) = \sum_{j=1}^{J} \beta_j \log(\mathbf{w}_j^T X X^T \mathbf{w}_j) + \beta_0 \qquad (5)$$

- for EEG-epoch $X \in \mathbb{R}^{CxD}$, estimate covariance ($\sim$power)
-
-
-
-

So what could this classifier look like?

We train and use the following classifier:

$$f(X, \mathbf{w}_1, ..., \mathbf{w}_J, \beta_0, ..., \beta_J) = \sum_{j=1}^{J} \beta_j \log(\mathbf{w}_j^T X X^T \mathbf{w}_j) + \beta_0 \qquad (5)$$

- for EEG-epoch $X \in \mathbb{R}^{CxD}$, estimate covariance ($\sim$power)
- filter the signal with CSP-filters $\mathbf{w}_1, ..., \mathbf{w}_J$
-
-
-

So what could this classifier look like?

We train and use the following classifier:

$$f(X, \mathbf{w}_1, ..., \mathbf{w}_J, \beta_0, ..., \beta_J) = \sum_{j=1}^{J} \beta_j \log(\mathbf{w}_j^T X X^T \mathbf{w}_j) + \beta_0 \qquad (5)$$

- for EEG-epoch $X \in \mathbb{R}^{CxD}$, estimate covariance ($\sim$power)
- filter the signal with CSP-filters $\mathbf{w}_1, ..., \mathbf{w}_J$
- take the logarithm of the power of the projected signal
-
-

So what could this classifier look like?

We train and use the following classifier:

$$f(X, \mathbf{w}_1, ..., \mathbf{w}_J, \beta_0, ..., \beta_J) = \sum_{j=1}^{J} \beta_j \log(\mathbf{w}_j^T X X^T \mathbf{w}_j) + \beta_0 \qquad (5)$$

- for EEG-epoch $X \in \mathbb{R}^{CxD}$, estimate covariance ($\sim$power)
- filter the signal with CSP-filters $\mathbf{w}_1, ..., \mathbf{w}_J$
- take the logarithm of the power of the projected signal
- $\beta_0, ..., \beta_J$ are the bias and weights , that need to be trained on sample epochs (e.g. via LDA)
-

So what could this classifier look like?

We train and use the following classifier:

$$f(X, \mathbf{w}_1, ..., \mathbf{w}_J, \beta_0, ..., \beta_J) = \sum_{j=1}^{J} \beta_j \log(\mathbf{w}_j^T X X^T \mathbf{w}_j) + \beta_0 \qquad (5)$$

- for EEG-epoch $X \in \mathbb{R}^{CxD}$, estimate covariance ($\sim$power)
- filter the signal with $\mathbf{w}_1, ..., \mathbf{w}_J$
- take the logarithm of the power of the projected signal
- $\beta_0, ..., \beta_J$ are the bias and weights, that need to be trained on sample epochs (e.g. via LDA)
- sign of the weighted sum is a prediction of labels (L/R labels need to be exchanged with $+1/-1$)