Machine-Learning for Brain Signal Analysis

Vincent Guigue
vincent.guigue@lip6.fr

September 9^{th} 2016

SMART Summer School

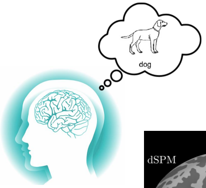
SORBONNE UNIVERSITÉS UPMC

- Which signals ?

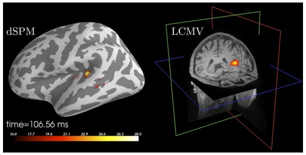
  [Non-invasive technologies]

  - EEG
  - MEG
  - fMRI

- Real-life issues ?
  - Medical diagnose
  - Brain understanding
    - Source localisation
    - Brain reading



- Machine-Learning issues ?
  - Classification
  - Regression
  - + Transfer
  - + Specific framework : 0-shot learning



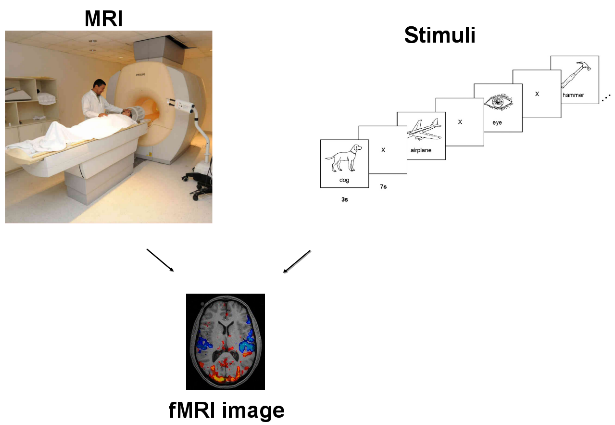
Raw data

⇒ Spatio-(temporal) data, sensor networks

⇒ Personalized signal

Non-invasive technologies

- fMRI



weak temporal aspect

Raw data

⇒ Spatio-(temporal) data, sensor networks

⇒ Personalized signal

Non-invasive technologies

- fMRI
- MEG

high noise level

Raw data

⇒ Spatio-(temporal) data, sensor networks

⇒ Personalized signal

Non-invasive technologies

- fMRI
- MEG
- EEG

Issues & machine learning approaches

<table>
<tr><th colspan="2">General problem</th><th>ML techniques</th><th>Specific settings</th></tr>
<tr><td rowspan="3">Signal classification</td><td>P300
BCI</td><td>Signal (pre-)Processing
Classifier (SVM, Ridge, LASSO)

Riemannian Geometry</td><td>Transfer learning</td></tr>
<tr><td>Seizure detection</td><td>Convolutional network (deep learning)</td><td></td></tr>
<tr><td>Brain Reading</td><td>Neural network
Latent representation</td><td>Transfer learning
0-shot learning</td></tr>
<tr><td colspan="2">Source localization</td><td>Regression</td><td>Inverse problem</td></tr>
</table>

Signal aquisition

Brain Computer Interface : P300-speller

Communication process

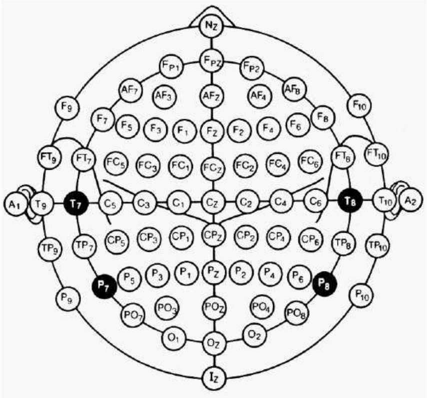
- Line/Column brief enlightenment = stimulus
- Brain response (300ms later)

Signal characteristics

- Good temporal resolution / bad spatial resolution
- High noise level...
- ... Require redundancy : aim = recognize the 30 positive samples among the $180 = 12 \times 15$ row and columns intensifications (**for one character**)

X-EG Datasets

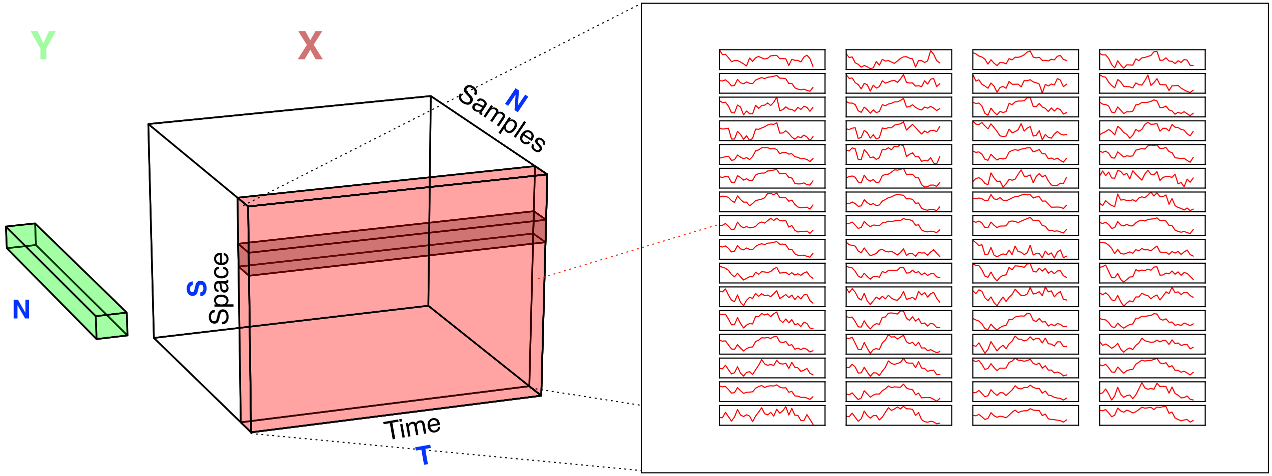
- **Spatial information :** sensors are placed according to standard patterns, e.g. :



EEG : 14 (epoc), 64 (usually), 118...
MEG : > 300, 2 kind of sensors

- **Temporal Information :** usual sampling $30Hz < f < 1000Hz$

Dataset & notations

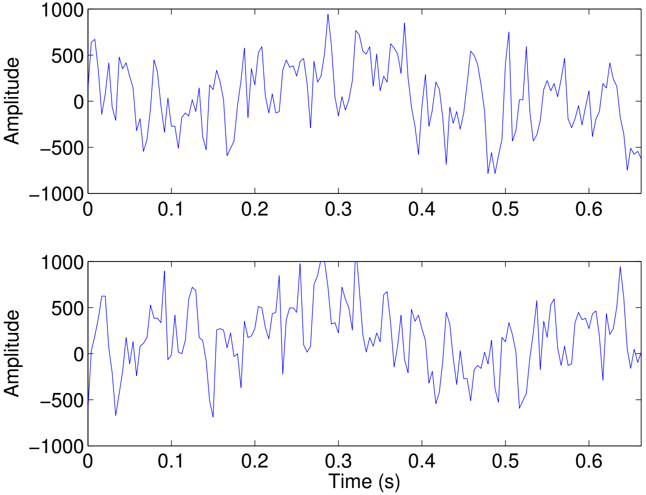


- **N** samples can be divided in **U** users
- Each user can be splitted in **Ns** sessions

Difficulties

P300 exemples :

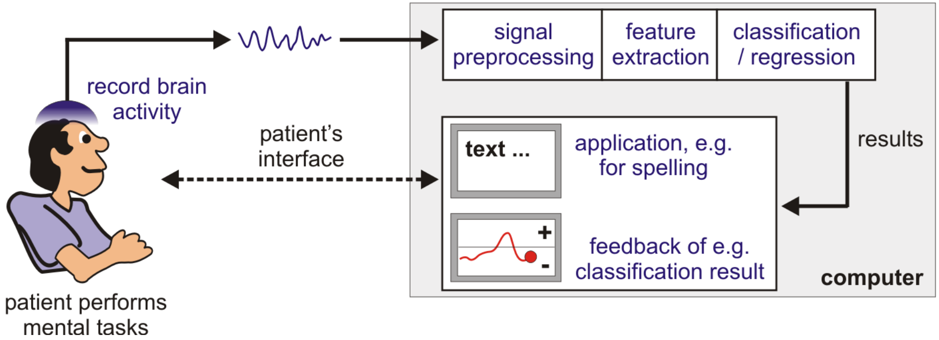
- Data from the BCI Competition 2003 provided by the Wadsworth Institute
- EEG acquisition : 64 Channels scalp sampled at 240 Hz
- Single user and 3 acquisition sessions spelling (5,6 and 8 words)



Positive & negative samples

Unbalanced dataset & (very) **high noise** level !
⇒ ML techniques are not able to tackle efficiently raw data (yet)

Processing chain



Crédit : M. Tangermann

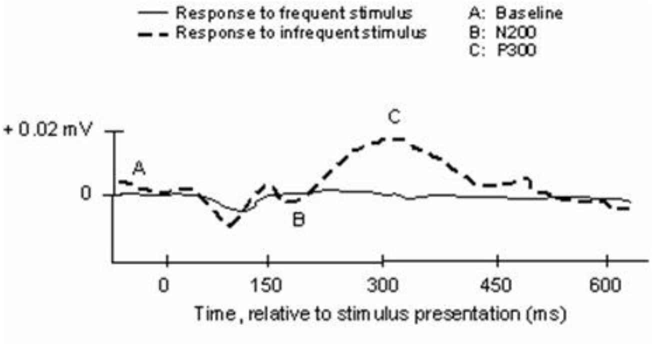
Step 1 : how to reduce the noise level ?

Sample aggregation

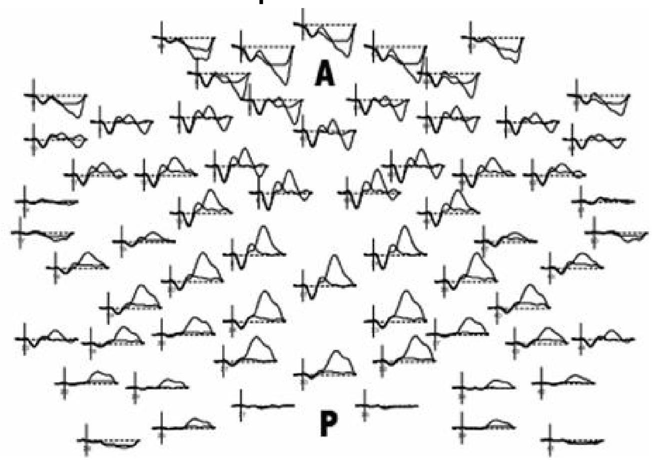
Credit : Patel and Azzam, 2005

Event Related Potential (mean of signals) :

Over one channel



Over the scalp



- A powerful tool to understand...
- ... Harder to classify single sample.

Patel and Azzam, 2005

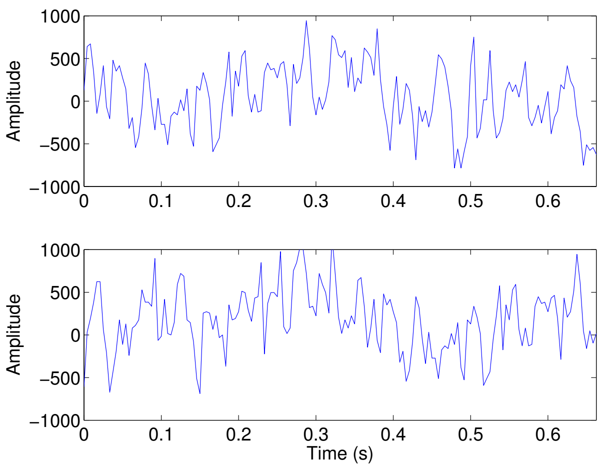
Characterization of N200 and P300 : Selected Studies of the Event-Related Potential

Filtering

- Phenomenon to catch = low frequency
- Noise ≈ high frequency

⇒ **Low pass filter**

- Extract 666-ms length signal after the intensification (P300 phenomena)
- Bandpass filtering and signal decimation : 0.1-20 Hz
- Each channel is composed of 14 samples

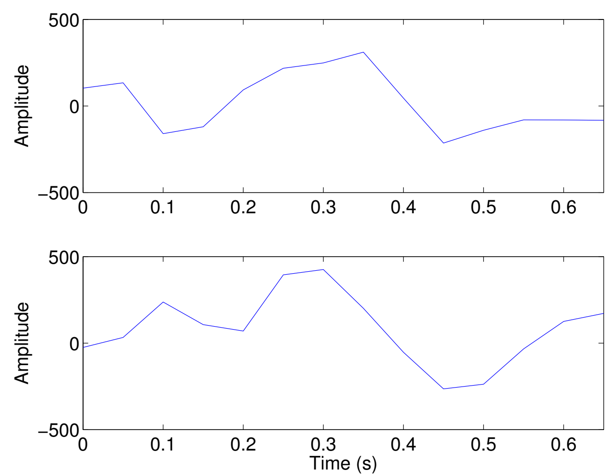


Filtering

- Phenomenon to catch = low frequency
- Noise ≈ high frequency

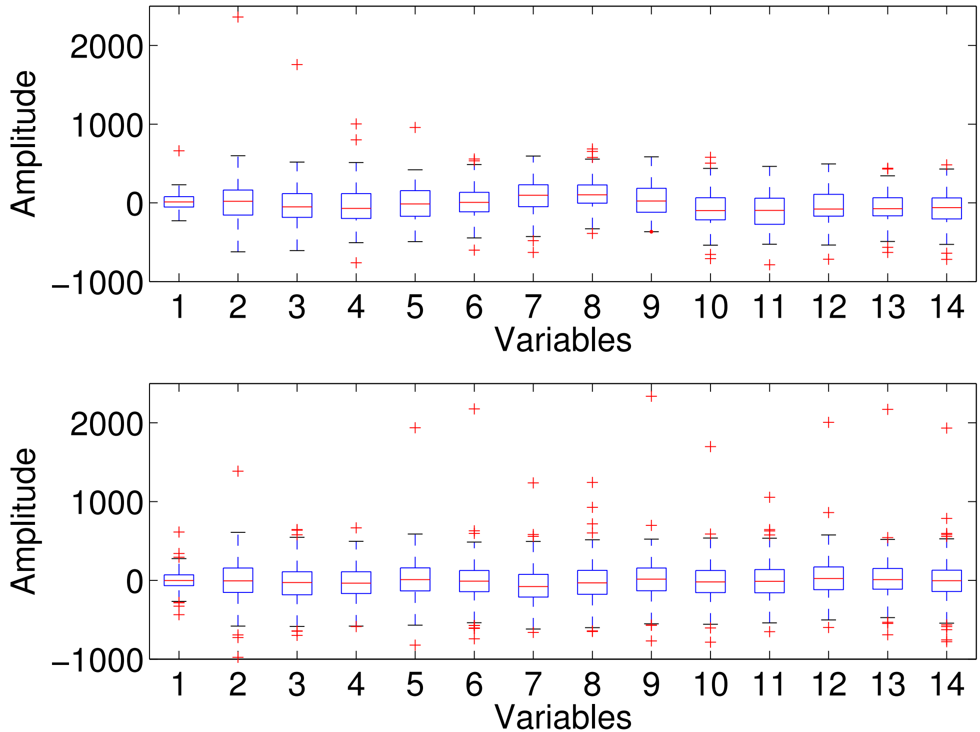
⇒ **Low pass filter**

- Extract 666-ms length signal after the intensification (P300 phenomena)
- Bandpass filtering and signal decimation : 0.1-20 Hz
- Each channel is composed of 14 samples



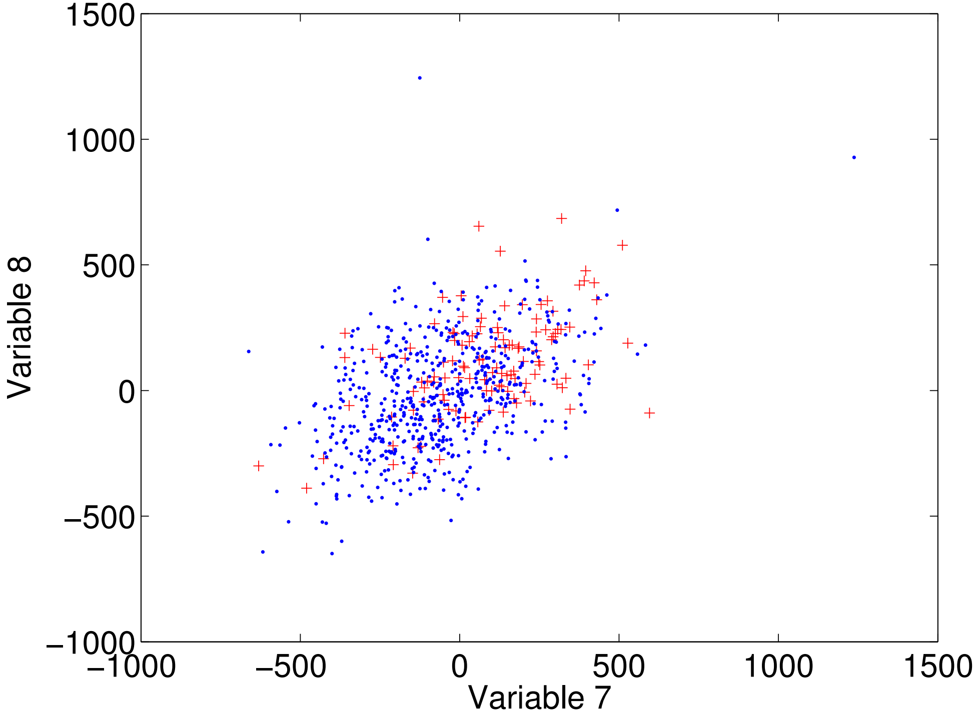
Filtering + sample aggregation

The problem remain difficult :



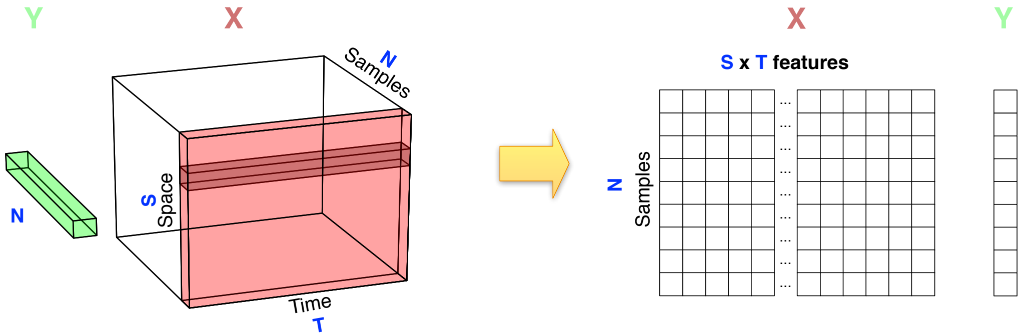
Filtering + sample aggregation

The problem remain difficult :



Plot of variable 7 *vs* variable 8 (≈ 300 ms)

Spatial aggregation : Concatenation

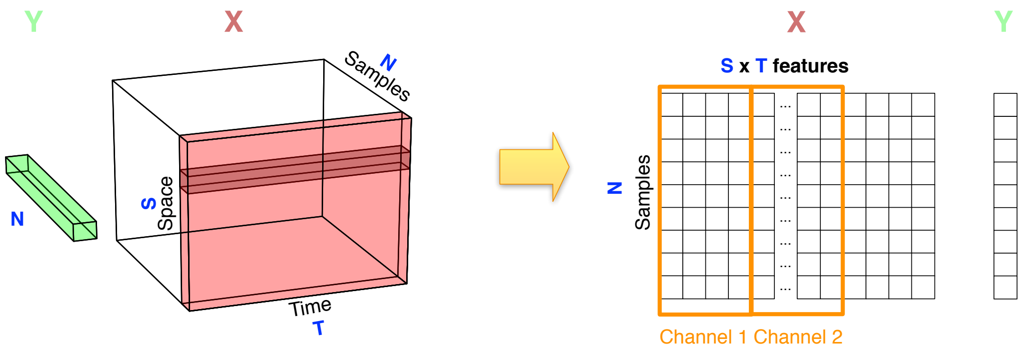


- Linear classifier :

$$f(\mathbf{x}_i) = \sum_j w_j x_{ij} \approx y_i$$

- No satisfactory performances

Spatial aggregation : Concatenation



- Linear classifier :

$$f(\mathbf{x}_i) = \sum_j w_j x_{ij} \approx y_i$$

- No satisfactory performances
- $\Rightarrow$ (bloc) feature selection : finding which channel are important... Or not. = eliminating bloc of **w**

Spatial sensor selection

Our solution to win *BCI Competition III : Dataset II* : **channel selection**

- filtering 0.1-20 Hz + signal decimation (14 measures / signal)
- post-stimulus signals coming from the spelling of a single word as training set
- Linear Support Vector Machines
- Feature selection by **Recursive Channel Elimination** with criterion

$$Crit = \frac{TP}{TP + FP + FN}$$

Intuition : select the subset of channels that maximizes this criterion

A. Rakotomamonjy, V. Guigue, 2008
BCI Competition III : Dataset II - Ensemble of SVMs for BCI P300 speller

Recursive Channel Elimination

A simple (& costly) approach :

Initialization : RANKED= $\emptyset$; CHANNEL= $[1, \cdots, d]$;
while *CHANNEL is not empty* **do**
  for *i in CHANNEL* **do**
    Remove temporarily channel i in CHANNEL;
    Learn a linear SVM with the remaining channel;
    Compute ranking criterion $Crit^{-(i)}$;
  end
  RANKCHAN= $\arg\min_i Crit^{-(i)}$;
  RANKED = [RANKCHAN RANKED];
  CHANNEL = CHANNEL / RANKCHAN ;
end

Algorithm 1: Variable ranking with backwards algorithm

Recursive Channel Elimination

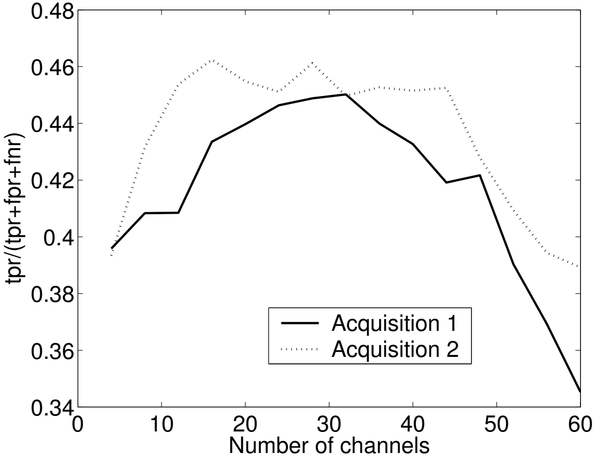
A simple (& costly) approach :

Initialization : RANKED$=\emptyset$; CHANNEL$=[1,\cdots,d]$;
while *CHANNEL is not empty* **do**
  for *i in CHANNEL* **do**
    Remove temporarily channel i in CHANNEL;
    Learn a linear SVM with the remaining channel;
    Compute ranking criterion $Crit^{-(i)}$;
  end
  RANKCHAN$=\arg\min_i Crit^{-(i)}$;
  RANKED = [RANKCHAN RANKED];
  CHANNEL = CHANNEL / RANKCHAN ;
end

Algorithm 2: Variable ranking with backwards algorithm

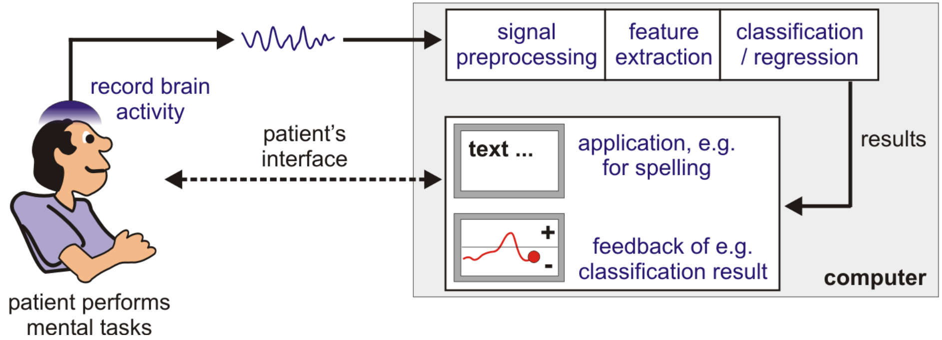
Feature Selection Results

- learning with 2 different sets lead to very different results
- best number of channels varies between 10 and 30
- performance varies between 0.35 and 0.46



Sessions	10 Top Ranked Channels									
1	9	15	18	36	40	55	56	59	63	64
2	18	39	53	55	56	58	59	60	61	64
3	9	18	40	48	53	55	56	58	61	64
4	10	18	33	42	46	55	56	58	60	64
5	16	22	39	40	50	56	57	60	61	62

Processing chain



Crédit : M. Tangermann

Step 2 : which classifier ?

Several alternatives (even for linear classifier)

Classical(& robust) linear classifier :

$$f(\mathbf{x}_i) = \sum_j w_j x_{ij} \approx y_i$$

- Logistic Regression (max likelihood)

$$\mathbf{w}^\star = \arg\max_{\mathbf{w}} \prod_i P(s_{\mathbf{w}}(\mathbf{x}_i) = 1|\mathbf{x}_i)^{y_i} \times [1 - P(s_{\mathbf{w}}(\mathbf{x}_i) = 1|x_i)]^{1-y_i}$$

- SVM (L1 cost, L2 regularization)
- LASSO (L2 cost, L1 regularization)
- Ridge regression (L2 cost, L2 regularization)

$$\mathbf{w}^\star = \arg\max_{\mathbf{w}} \sum_i \Delta(f_{\mathbf{w}}(\mathbf{x}_i), y_i) + \lambda\Omega(\mathbf{w})$$

No impact in our chain... But many opportunities in other contexts.

Single classifier *vs* Ensemble of classifiers

- High signal variability $\Rightarrow$ require robust classifier
- Single classifier fails...
- ... Ensemble of classifiers succeed !

Using ensemble of classifiers...

- is a way to robustify statistical decision
- & became a basic rule to obtain good Kaggle performances

$\Rightarrow$ Require a way to merge outputs.

Merging classifiers

Each classifier is trained on a word (sessions contain resp. 5, 6 and 8 words).

How to recognize a character from the 15 sequences ?

- Let $\mathbf{x}_i$ be post-stimulus signal associated to the illumination of a row or a column
- Each classifier scores bx_i through $f_k(bx_i)$
- Update the overall score of the given row/column

$$S_{rc} = S_{rc} + \sum_k f_k(bx_i)$$

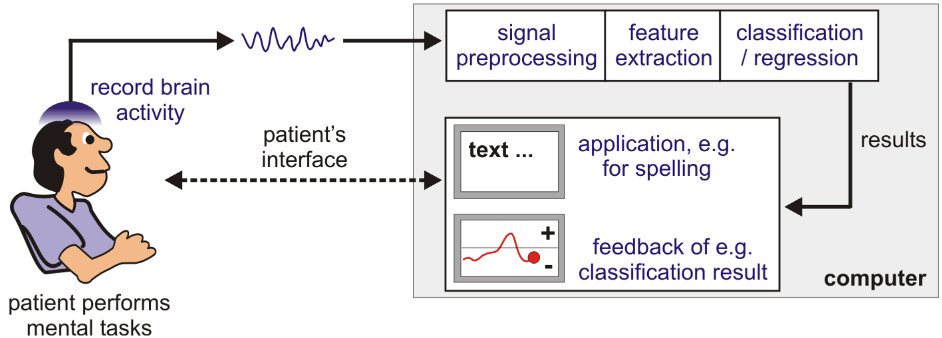
- After all the sequences, select the character which corresponds to the highest row and columns scores.

Characters Spelling Results

	Nb. of sequences							
Algorithms	1	2	3	4	5	6	7	10
10 preselected channels and single SVM	14	6	6	0	1	0	0	0
all channels and single SVM	14	10	9	5	5	5	1	0
10 preselected channels and Ens. SVM	13	8	3	1	2	0	0	0
all channels and Ens. SVM	7	4	3	0	0	0	0	0
4 relevant channels and Ens. SVM	8	7	4	0	1	0	0	0
10 relevant channels and Ens. SVM	8	5	5	1	0	1	0	0
26 relevant channels and Ens. SVM	4	2	0	0	0	0	0	0
30 relevant channels and Ens. SVM	5	3	0	0	0	0	0	0
optimal relevant channels and Ens. SVM	4	2	1	0	0	0	0	0

TABLE: Errors wrt the nb of illumination sequences

Processing chain



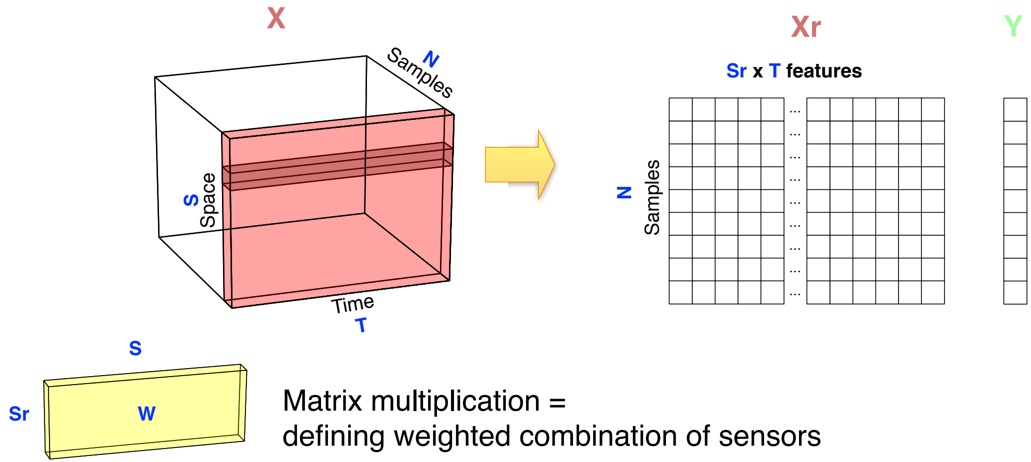
Crédit : M. Tangermann

Which alternatives ?

Can we merge pre-processing & training steps ?

(≈) New issues in ML techniques for EEG analysis

CSP : Common Spatial Pattern

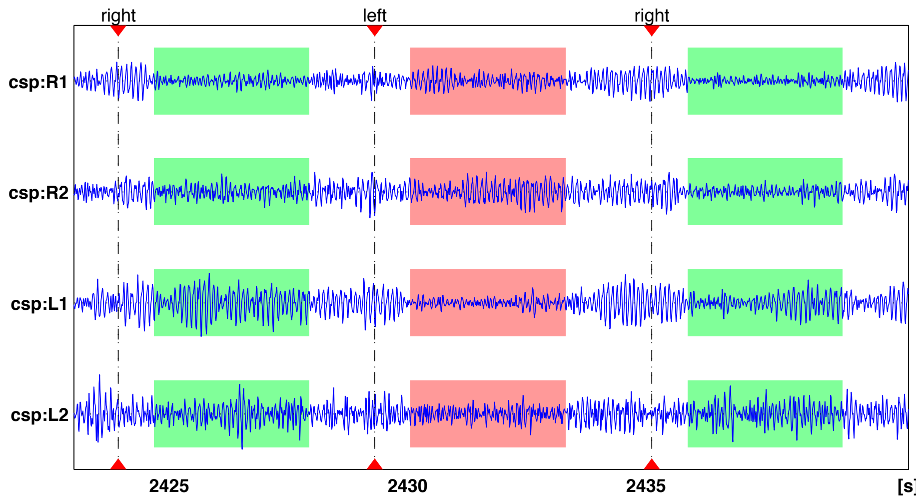


- Orthogonal sensor combinations maximizing the variance ($\approx$ PCA in sensor space)
- Combining sensor = noise reduction

ZJ Koles, MS Lazar, SZ Zhou, 1990
Spatial patterns underlying population differences in the background EEG

CSP : one of the key of Robust EEG Single-Trial Analysis

Exemples of use in motor cortex imagery

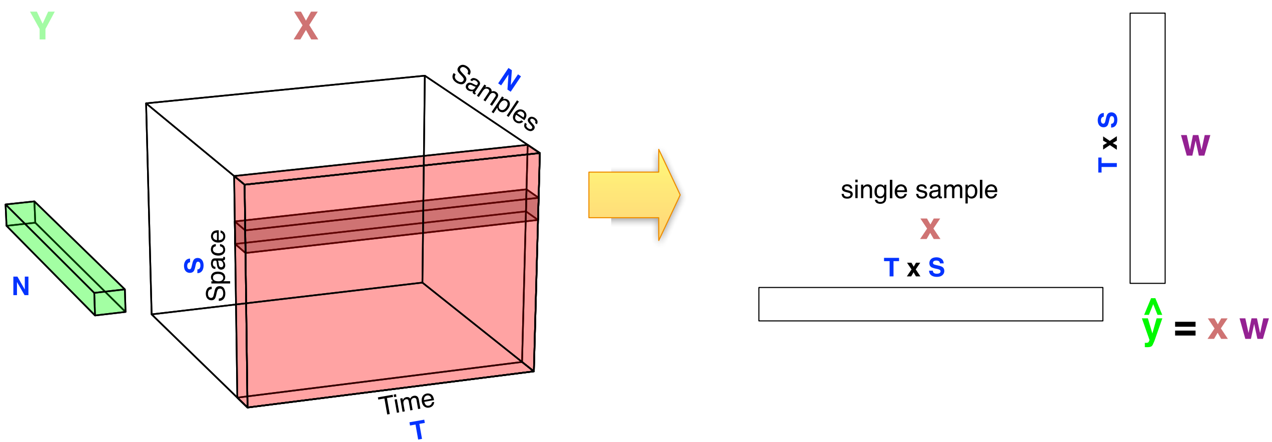


Left *vs* right hand move mapped to 4 aggregated channels.

Blankertz et al., 2008
Optimizing Spatial Filters for Robust EEG Single-Trial Analysis

Bi-linear SVM

- Using a linear classifier = losing structure information



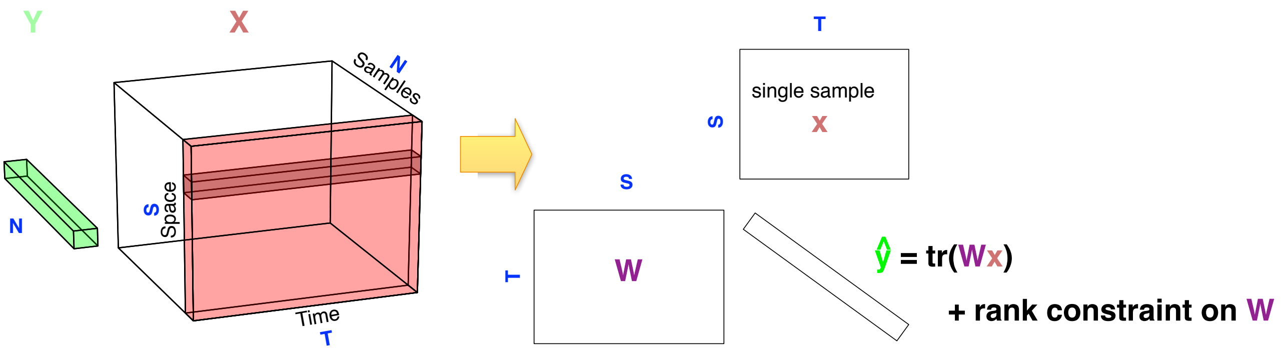
How can we impose structural constraints on **w** ? ?

Pirsiavash et al., NIPS 2009

Bilinear classifiers for visual recognition

Bi-linear SVM

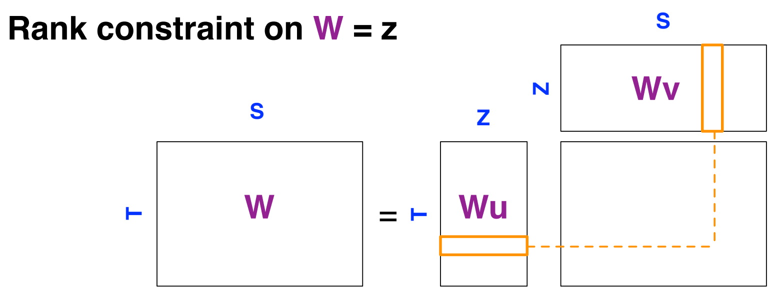
- Using a linear classifier = losing structure information
- bilinear classifiers ⇒ Modeling variable dependencies on 2 axis (time/space)



Pirsiavash et al., NIPS 2009
Bilinear classifiers for visual recognition

Bi-linear SVM

- Using a linear classifier = losing structure information
- bilinear classifiers ⇒ Modeling variable dependencies on 2 axis (time/space)

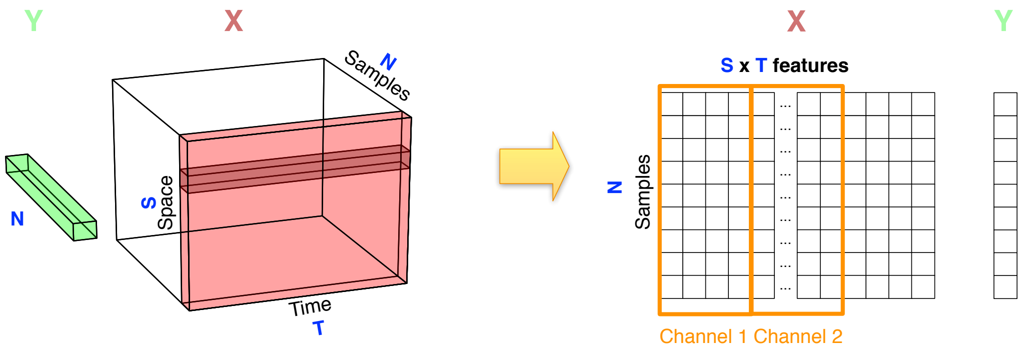


⇒ Structural consistency in the way of building W

Pirsiavash et al., NIPS 2009

Bilinear classifiers for visual recognition

Regularization as a selection procedure with linear classifiers



$$f(\mathbf{x}_i) = \sum_j w_j x_{ij} \approx y_i$$

General training formulation :

$$\mathbf{W}^\star = \arg\max_{\mathbf{W}} \sum_i \Delta(f_{\mathbf{W}}(\mathbf{x}_i), y_i) + \lambda \Omega(\mathbf{W}), \qquad \mathbf{W}^\star \in \mathbb{R}^{S \times T}$$

Regularization as a selection procedure with linear classifiers (2)

$$\mathbf{W}^\star = \arg\max_{\mathbf{W}} \sum_i \Delta(f_{\mathbf{W}}(\mathbf{x}_i), y_i) + \lambda\Omega(\mathbf{W}), \qquad \mathbf{W}^\star \in \mathbb{R}^{S\times T}$$

Regularization as a selection procedure with linear classifiers (2)

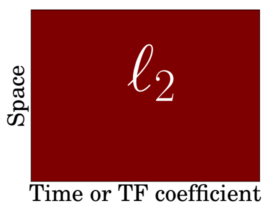
$$\mathbf{W}^\star = \arg\max_{\mathbf{W}} \sum_i \Delta(f_{\mathbf{W}}(\mathbf{x}_i), y_i) + \lambda\Omega(\mathbf{W}), \qquad \mathbf{W}^\star \in \mathbb{R}^{S\times T}$$

- **L2 regularization :** $\Omega(\mathbf{W}) = \sum_{j,k} w_{jk}^2$

  Associated update in a gradient descent procedure :

$$w_{jk} \leftarrow w_{jk} - 2\epsilon w_{jk} \Leftrightarrow w_{jk} \leftarrow w_{jk}(1-2\epsilon)$$

$\mathbf{W} =$



[credit Gramfort]

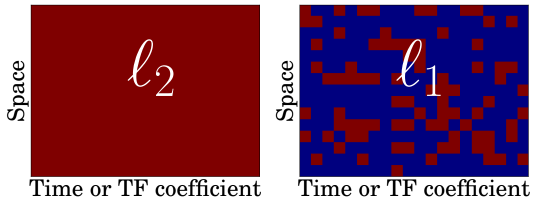
Regularization as a selection procedure with linear classifiers (2)

$$\mathbf{W}^\star = \arg\max_{\mathbf{W}} \sum_i \Delta(f_{\mathbf{W}}(\mathbf{x}_i), y_i) + \lambda\Omega(\mathbf{W}), \qquad \mathbf{W}^\star \in \mathbb{R}^{S\times T}$$

- **L1 regularization :** $\Omega(\mathbf{W}) = \sum_{j,k} |w_{jk}|$

  Associated update in a gradient descent procedure = **soft-thresholding** :

$$w_{jk} \leftarrow \begin{cases} w_{jk} - \epsilon \,\mathrm{sign}(w_{jk}) & \text{if } |w_{jk}| > \epsilon \\ 0 & \text{else} \end{cases}$$

$\mathbf{W} =$ 

[credit Gramfort]

Regularization as a selection procedure with linear classifiers (2)

$$\mathbf{W}^\star = \arg\max_{\mathbf{W}} \sum_i \Delta(f_{\mathbf{W}}(\mathbf{x}_i), y_i) + \lambda\Omega(\mathbf{W}), \qquad \mathbf{W}^\star \in \mathbb{R}^{S\times T}$$

Elastic net variant combines L1 and L2 for more stability

- Sparseness of L1,
- Robustness of L2

Zou, Hastie, 2005
Regularization and variable selection via the elastic net

Regularization as a selection procedure with linear classifiers (2)

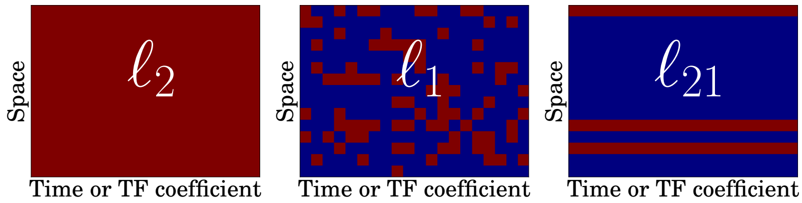
$$\mathbf{W}^\star = \arg\max_{\mathbf{W}} \sum_i \Delta(f_{\mathbf{W}}(\mathbf{x}_i), y_i) + \lambda\Omega(\mathbf{W}), \qquad \mathbf{W}^\star \in \mathbb{R}^{S\times T}$$

- **L21 regularization :** $\Omega(\mathbf{W}) = \sum_j \sqrt{\sum_k w_{jk}^2} = \sum_j \|\mathbf{w}_j\|$

  Sparsity at the sensor level Gradient descent update :

$$w_{jk} \leftarrow \begin{cases} w_{jk}(1 - \frac{\epsilon}{\|\mathbf{w}_j\|}) & \text{if } \|\mathbf{w}_j\| > \epsilon \\ 0 & \text{else} \end{cases}$$

$\mathbf{W} =$



[credit Gramfort]

G. Obozinski, B. Taskar, and M. I. Jordan, 2006
Multi-task feature selection

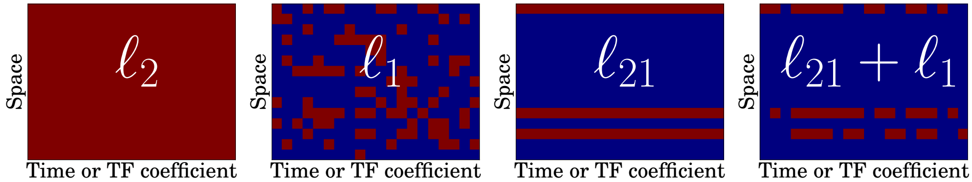
Regularization as a selection procedure with linear classifiers (2)

$$\mathbf{W}^\star = \arg\max_{\mathbf{W}} \sum_i \Delta(f_{\mathbf{W}}(\mathbf{x}_i), y_i) + \lambda\Omega(\mathbf{W}), \qquad \mathbf{W}^\star \in \mathbb{R}^{S\times T}$$

- **L21 regularization + L1 :**

  Playing with advanced (and dedicated formulation)

$\mathbf{W} =$



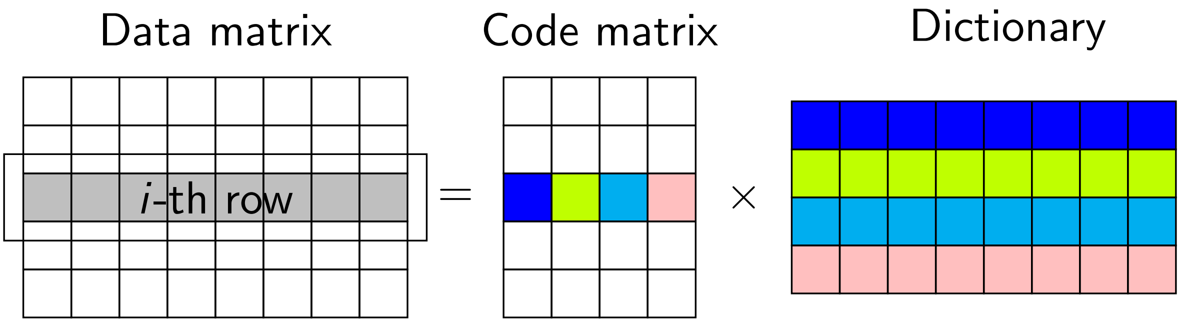
[credit Gramfort]

Gramfort et al., 2013
Time-Frequency Mixed-Norm Estimates : Sparse M/EEG imaging with non-stationary source activations

Representation learning / dictionary learning

Raw signal are very difficult to handle...

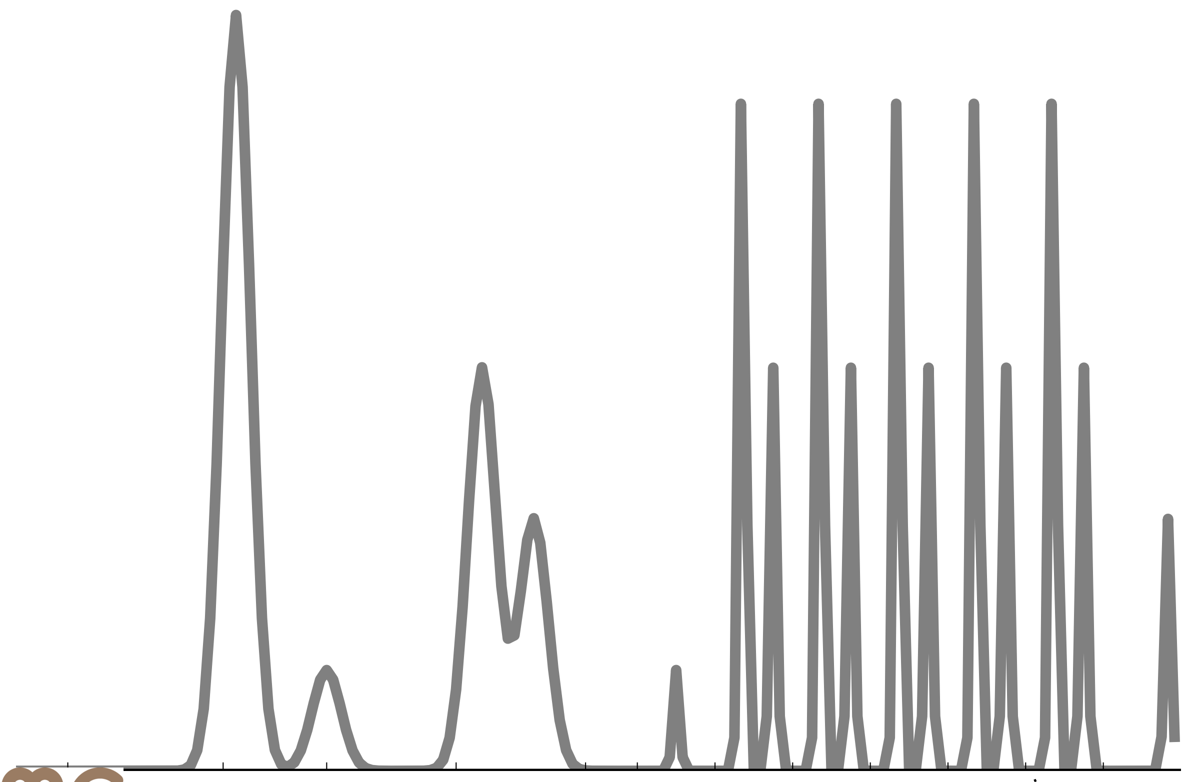
... Let learn a new space where the problem is easy to solve !



- Variations SVD [Golub 96].
  - Non-negative matrix factorization [Lee 2000]
  - Sparseness [Hoyer 2002]
- Learning criterion = reconstruction error
- Easy constraint design (to adapt to specific problems)
- Efficient solvers

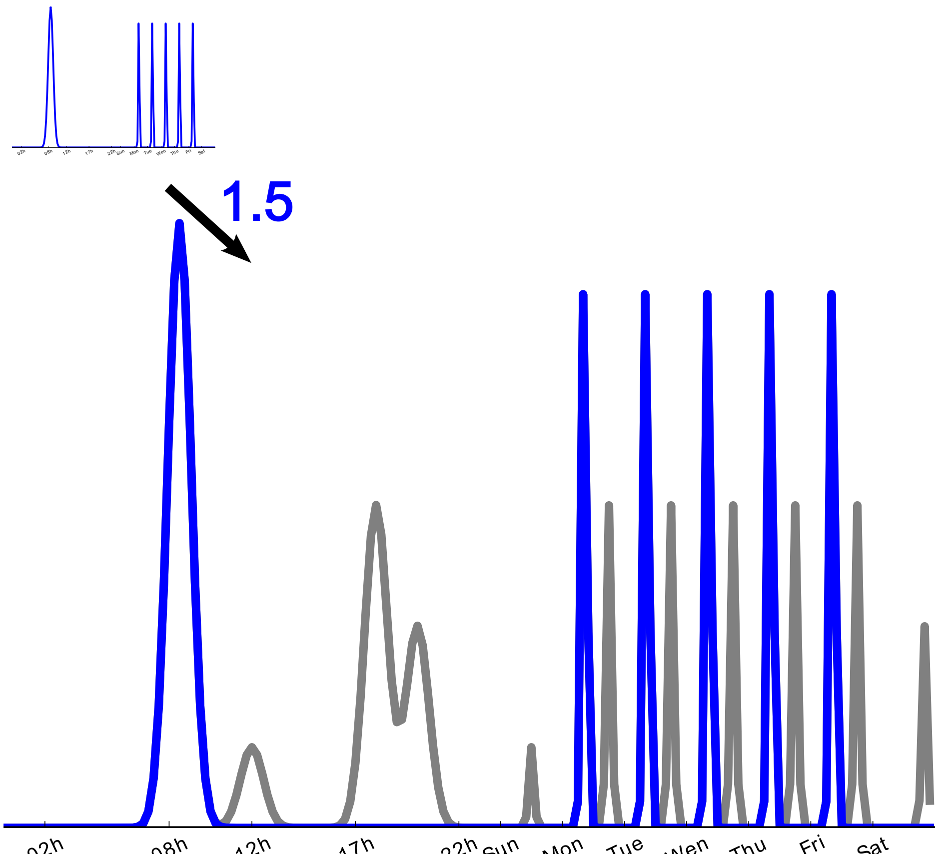
Representation learning / dictionary learning

With an exemple (far away from EEG...)



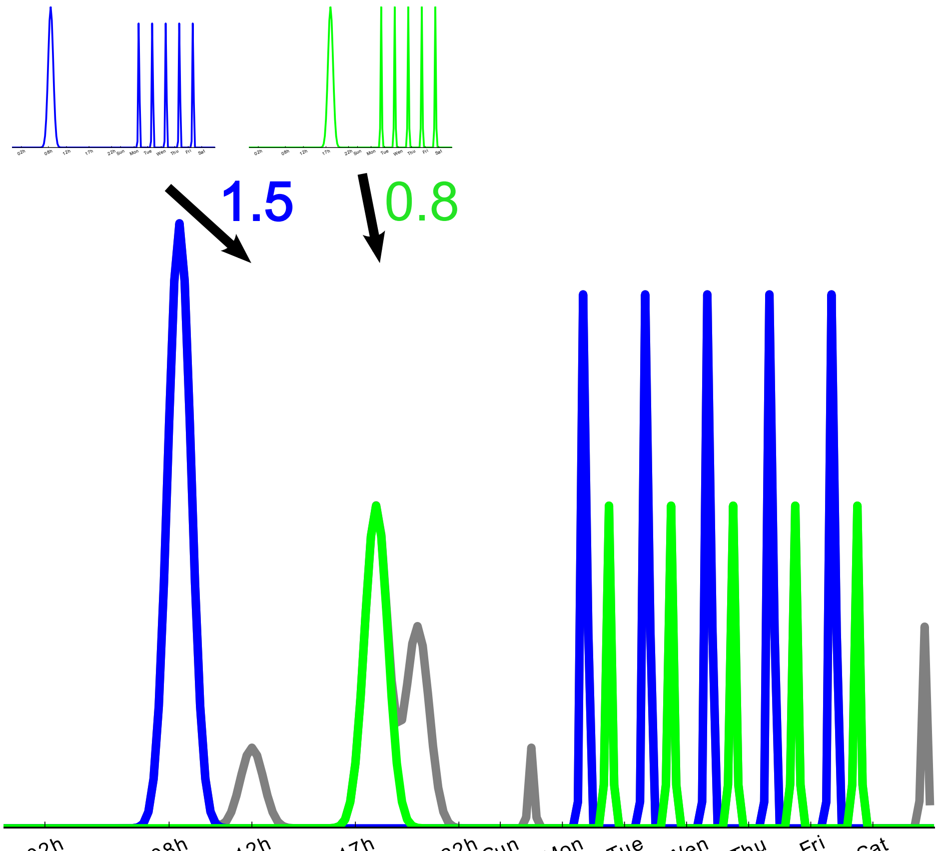
Representation learning / dictionary learning

With an exemple (far away from EEG...)



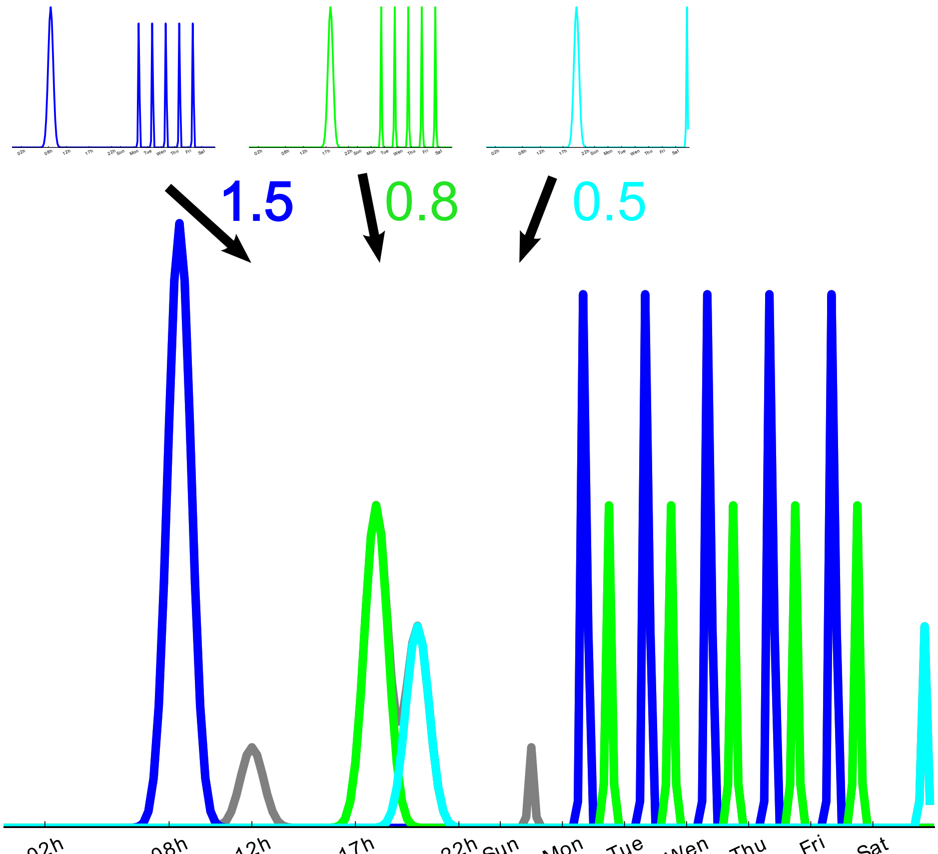
Representation learning / dictionary learning

With an exemple (far away from EEG...)



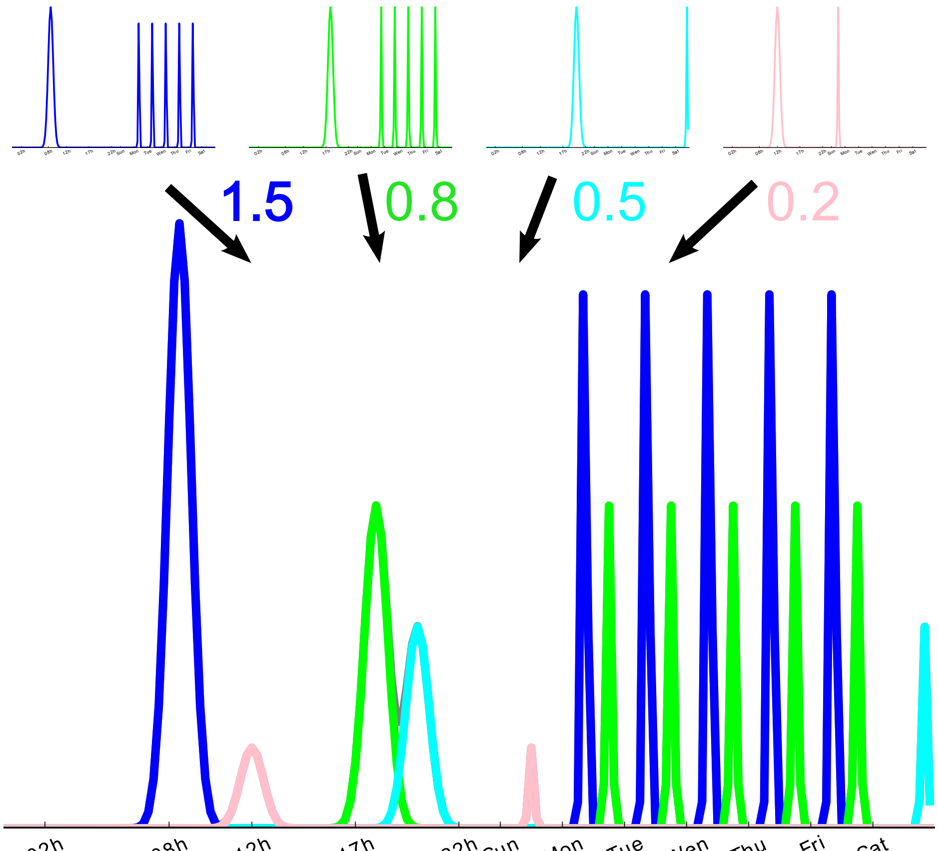
Representation learning / dictionary learning

With an exemple (far away from EEG...)



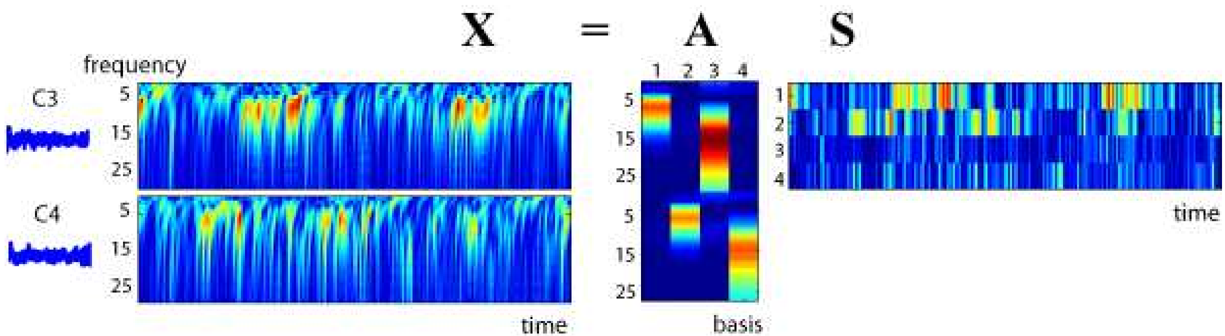
Representation learning / dictionary learning

With an exemple (far away from EEG...)



NMF and EEG

Extracting common pattern in time-frequency representation of EEG :



- Adding extra-constraints
- Gain when classifying A instead of X on BCI Challenge III (motor imagery)

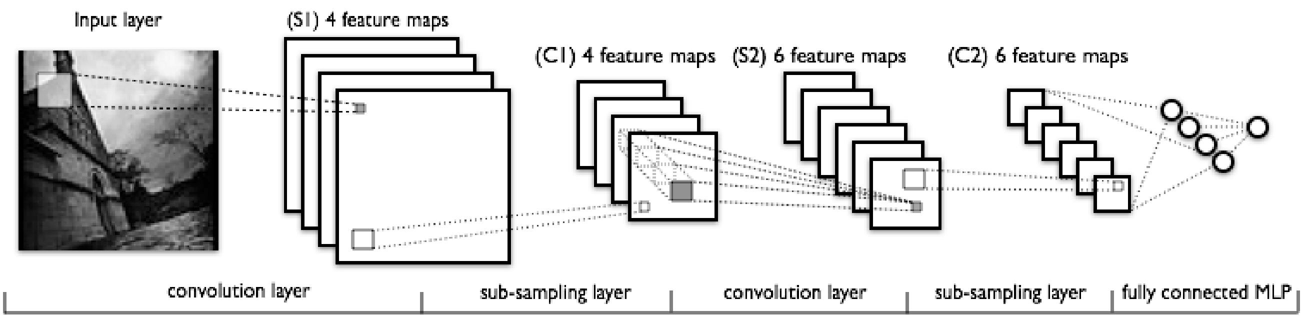
Lee and Choi, AISTATS 2009
Group Nonnegative Matrix Factorization for EEG Classification

Deep learning & EEG

Neural networks opportunities for EEG

- Extracting auto-learned features
- Modeling invariances (both time/space)

General CNN architecture



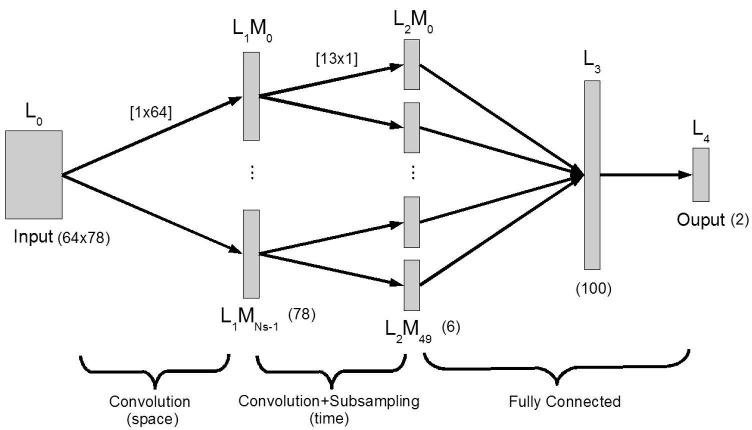
- Very efficient on many problems... But not so robust to noise
- Easy to understand... But hard to implement

Deep learning & EEG

Neural networks opportunities for EEG

- Extracting auto-learned features
- Modeling invariances (both time/space)

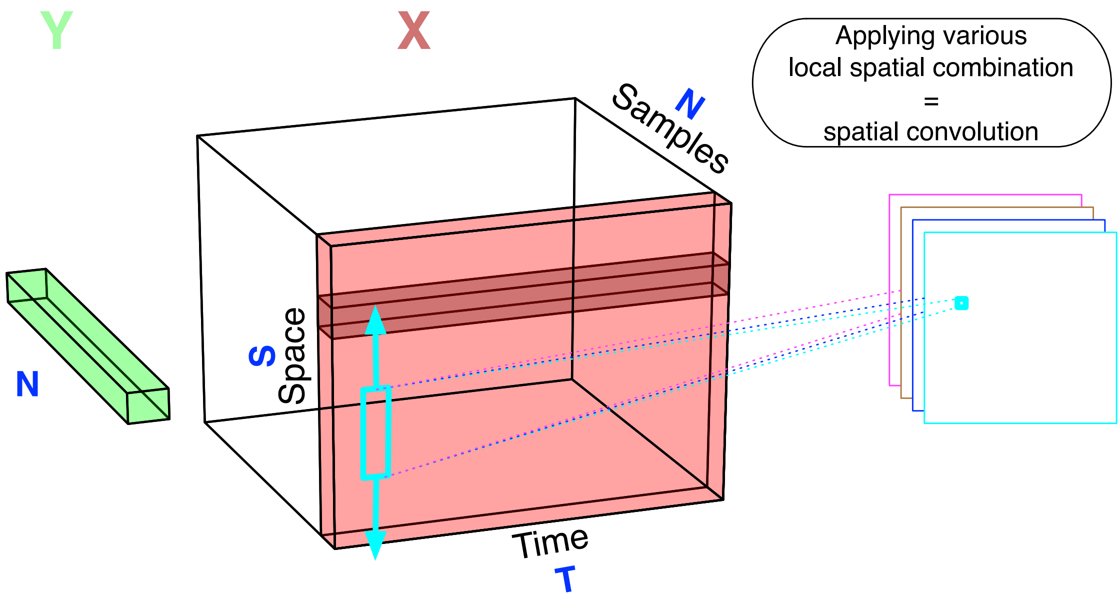
Cecotti Architecture dedicated to P300 :



Cecotti and Gräser, PAMI 2011
Convolutional Neural Networks for P300 Detection with Application to Brain-Computer Interfaces

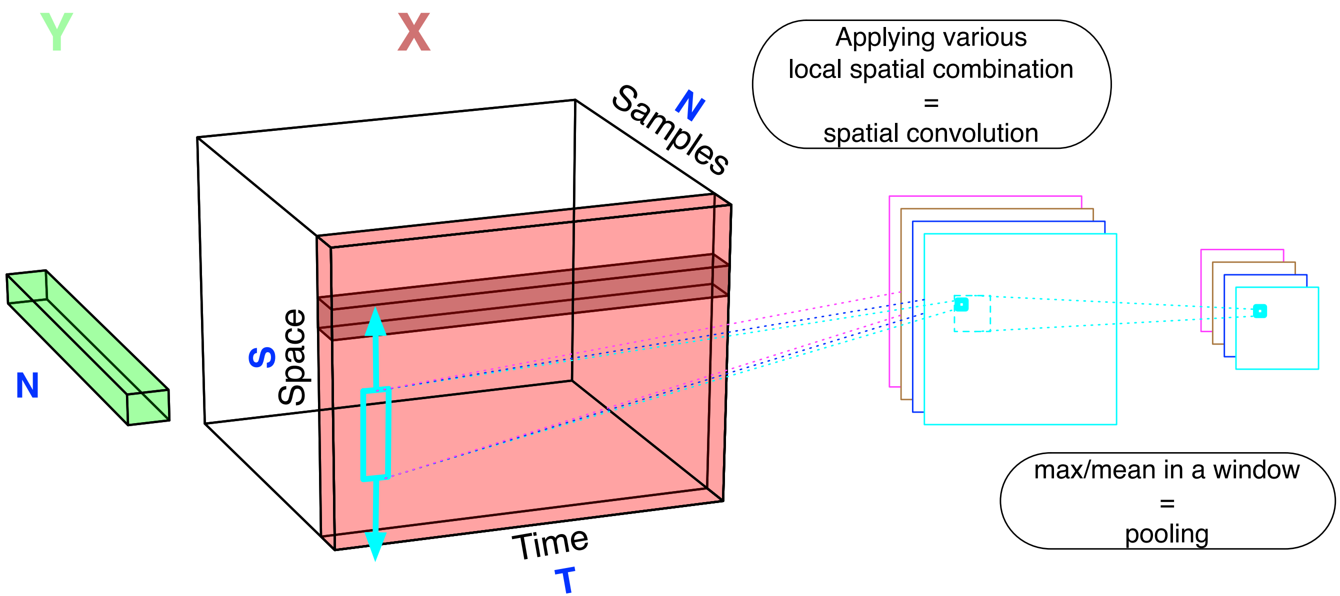
Deep learning & EEG

Detail of the architecture :



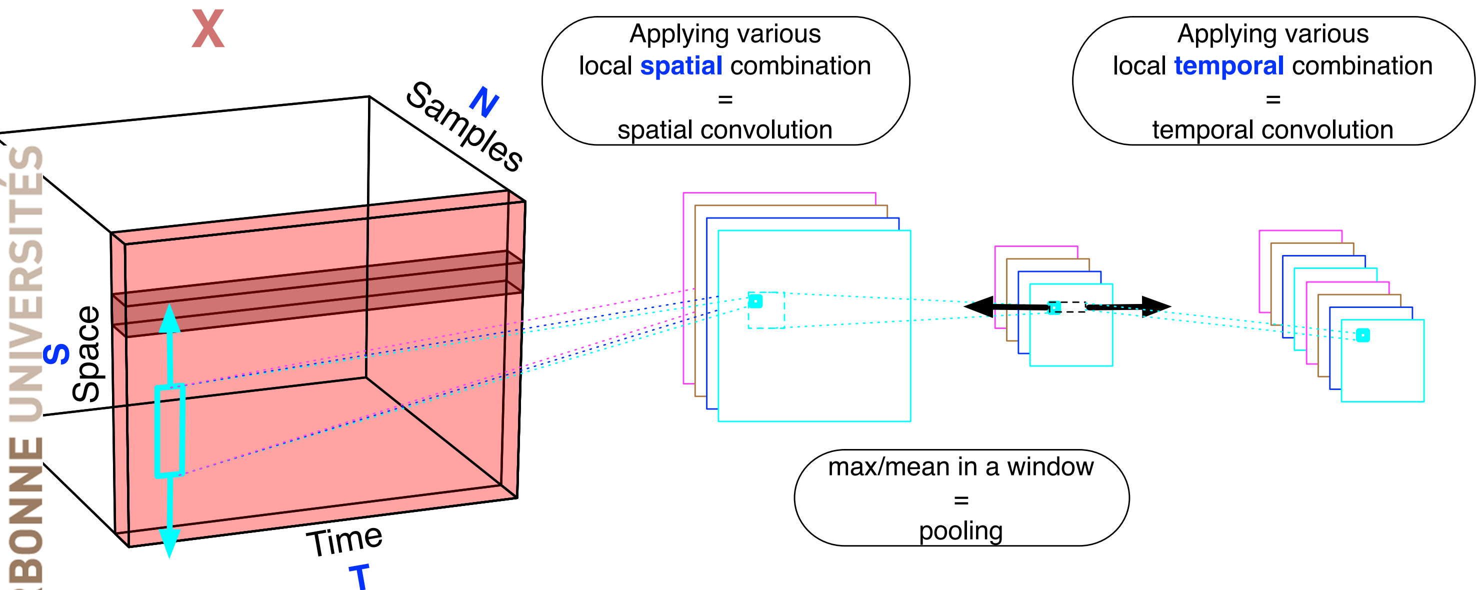
Deep learning & EEG

Detail of the architecture :



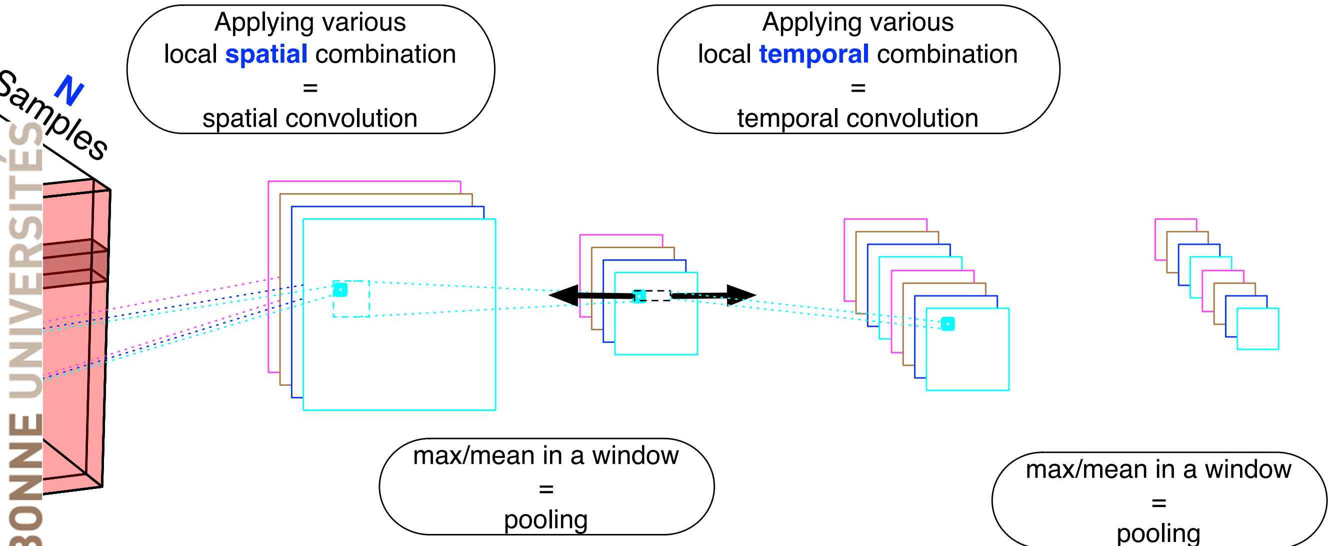
Deep learning & EEG

Detail of the architecture :



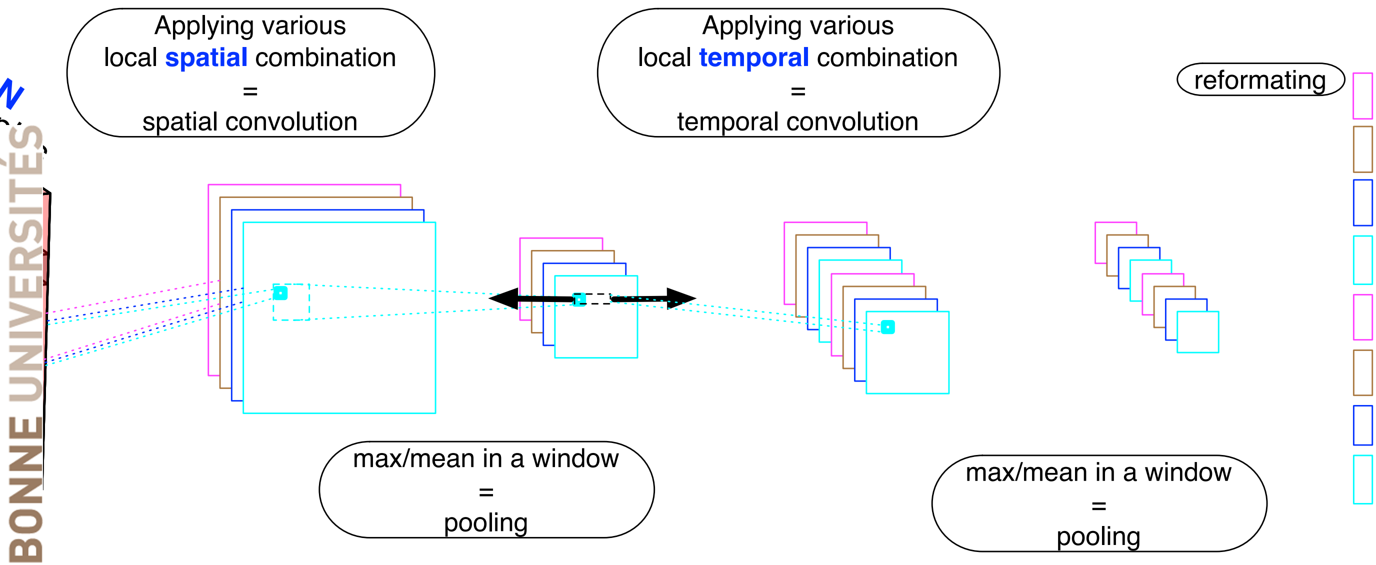
Deep learning & EEG

Detail of the architecture :



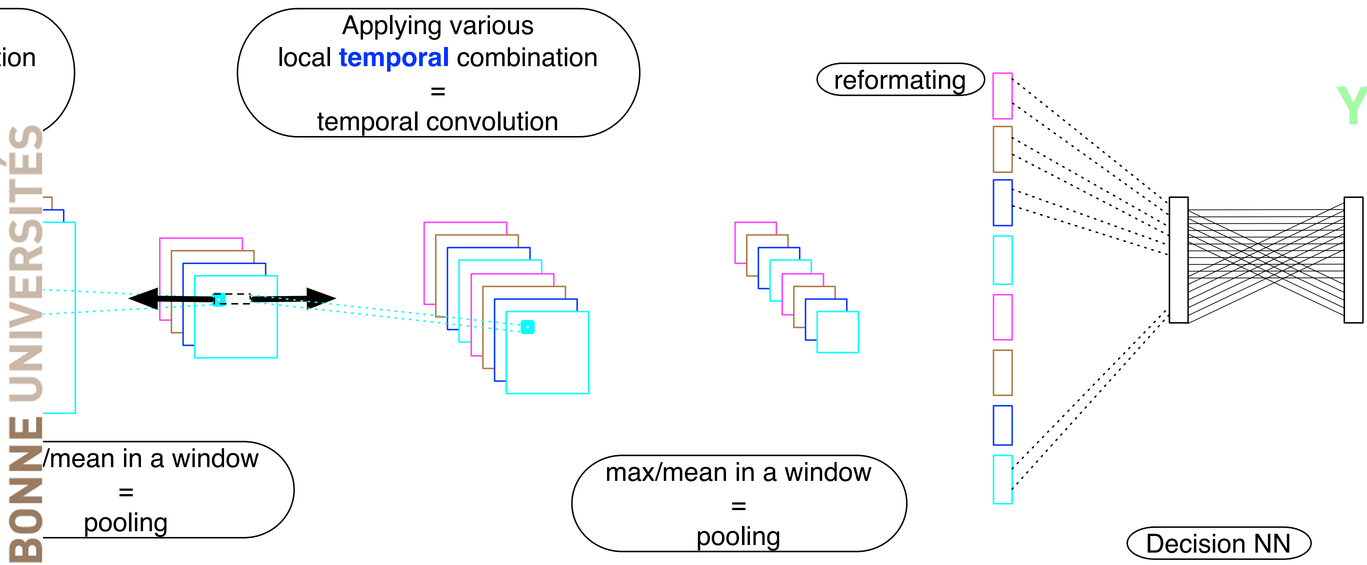
Deep learning & EEG

Detail of the architecture :



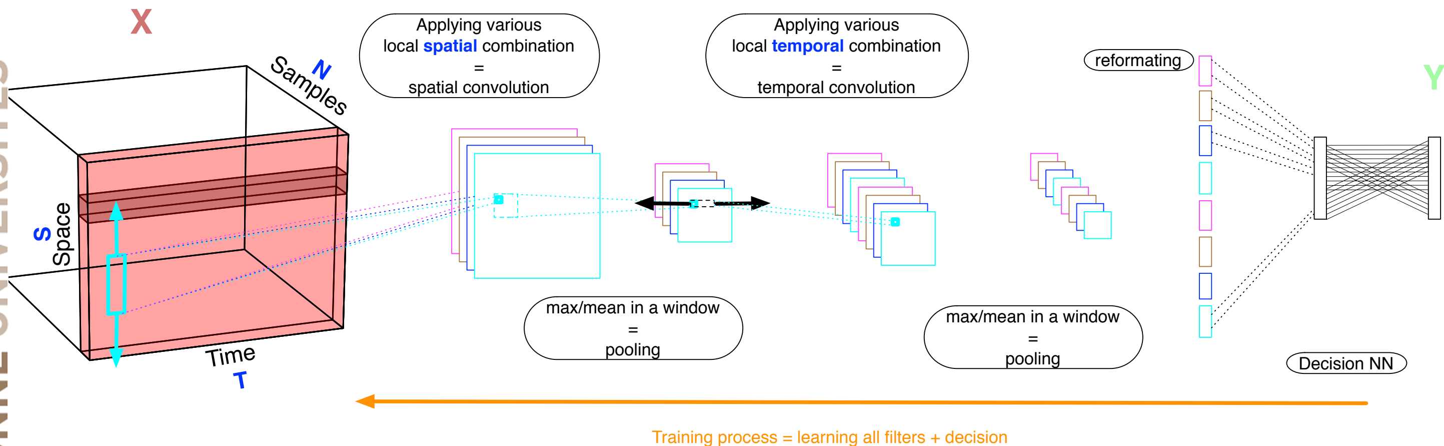
Deep learning & EEG

Detail of the architecture :



Deep learning & EEG

Detail of the architecture :



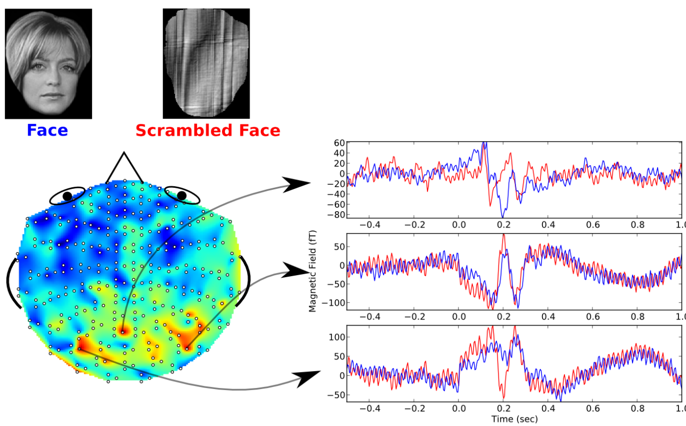
Transfer Learning

Our aim :

1. Training models on existing EEG dataset
2. Testing algorithms on new subjects

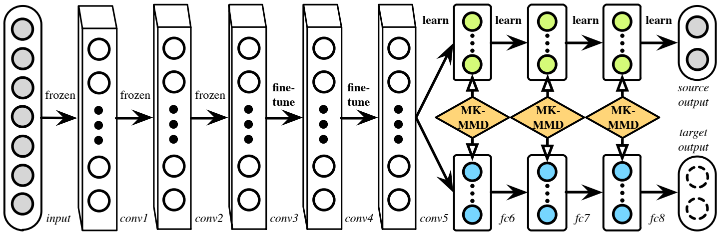
$\Rightarrow$ classical algorithms fail !

Kaggle MEG 2014 : *train* = 16 subjects ; *test* = 6 **different** subjects



Transfer Learning : which solutions ?

- Learning many classifier adapted to various topologies + aggregation/vote [easy]
- Extracting subject invariant features
  - NMF + constraints
  - Structural Correspondence Learning
  - NN + constraints on hidden layers



Blitzer et al., 2006
Domain adaptation with structural correspondence learning

Long et al., 2015
Learning Transferable Features with Deep Adaptation Networks

- Aligning data/classifiers from one patient to another

Transfer Learning : which solutions ?

- Learning many classifier adapted to various topologies + aggregation/vote [easy]
- Extracting subject invariant features
- Aligning data/classifiers from one patient to another
  - Iterative Procrustean alignment + classifier in a *universal* space

    Solving : $\min_T ||VT - L||$ with :

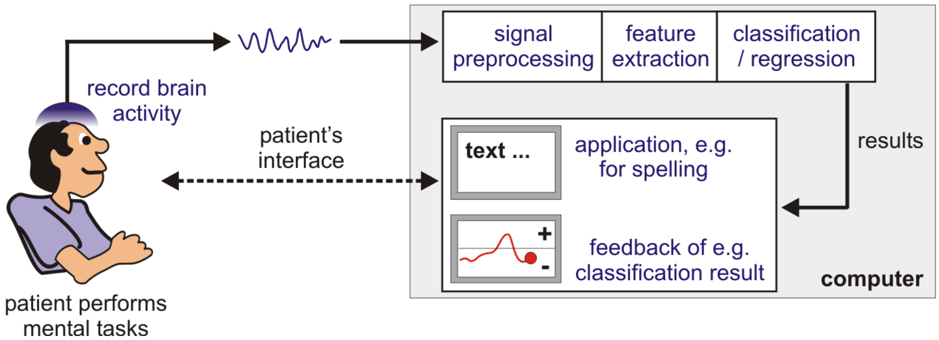
    $V \in \mathbb{R}^{k \times n}$ data to align

    $L \in \mathbb{R}^{k \times n}$ well known reference

    $T \in \mathbb{R}^{n \times n}$ transfer matrix

Haxby et al., 2011
A Common, High-Dimensional Model of the Representational Space in Human Ventral Temporal Cortex

Alexandre Barachant



Crédit : M. Tangermann

A real breakthrough for EEG classification...

... And transfer !

Winner of Kaggle competition MEG 2014, EEG 2015

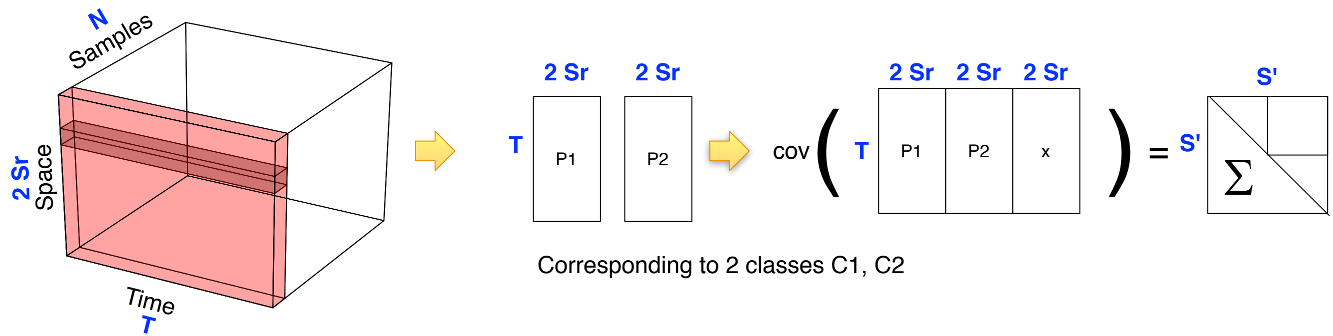
Alexandre Barachant approach

- Classical preprocessing filtering + CSP
- Feature = correlation on aggregated matrix (prototype + signal)
- Specific metrics in this new space

Butterworth filter
CSP (class by class) => 2 Sr channels

Averaging by class:
prototypes P1, P2

Processing of signal x:



Corresponding to 2 classes C1, C2

How handling those new data ?

Σ is SPD (semi positive definite) $\Rightarrow$ with Riemannian geometry

Basics in Riemannian geometry

- Distance between 2 samples :

$$\delta(\Sigma_1, \Sigma_2) = \| \operatorname{Log}(\Sigma_1^{-\frac{1}{2}} \Sigma_2 \Sigma_1^{-\frac{1}{2}}) \|_F$$

- Mean computation :

$$\Sigma^\star = \operatorname{mean}(\Sigma_1, \ldots, \Sigma_N) = \arg\min_{\Sigma} \sum_i \delta^2(\Sigma, \Sigma_i)$$

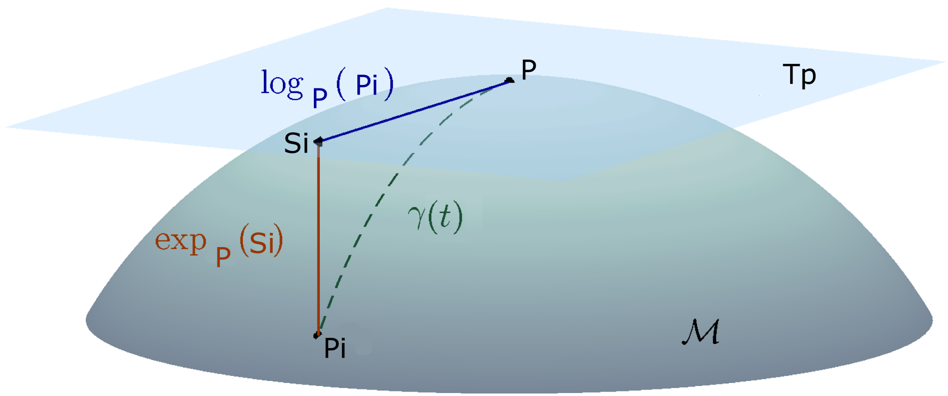
Simple idea :

1. Build a prototype corresponding to **each class** : $\Sigma^\star_{cl}$
2. Inférence on Σ : $C^\star = \arg\min_c \delta(\Sigma, \Sigma^\star_{cl})$

... But how computing $\Sigma^\star$?

Computing prototypes (=tangent point)

- $\mathcal{M}$: manifold of Σ objects ; $\mathcal{T}$: tangent space
- Mapping $\mathcal{M} \to \mathcal{T}$: $S = \phi_P(\Sigma) = P^{\frac{1}{2}} \operatorname{Log}(P^{-\frac{1}{2}} \Sigma P^{-\frac{1}{2}}) P^{\frac{1}{2}}$
- Inverse mapping : $\Sigma = \phi_P^{-1}(S) = P^{\frac{1}{2}} \operatorname{Exp}(P^{-\frac{1}{2}} S P^{-\frac{1}{2}}) P^{\frac{1}{2}}$



[Barachant]

Computing prototypes (=tangent point)

- $\mathcal{M}$: manifold of Σ objects ; $\mathcal{T}$: tangent space
- Mapping $\mathcal{M} \to \mathcal{T}$: $S = \phi_P(\Sigma) = P^{\frac{1}{2}} \operatorname{Log}(P^{-\frac{1}{2}} \Sigma P^{-\frac{1}{2}}) P^{\frac{1}{2}}$
- Inverse mapping : $\Sigma = \phi_P^{-1}(S) = P^{\frac{1}{2}} \operatorname{Exp}(P^{-\frac{1}{2}} S P^{-\frac{1}{2}}) P^{\frac{1}{2}}$

Algorithm :

1. Init : $P = \dfrac{1}{N} \displaystyle\sum_{i=1}^{N} \Sigma_i$
2. while $\|S\|_F > \epsilon$:
$$S = \frac{1}{N} \sum_i \phi_P(\Sigma_i)$$
$$P = \phi_P^{-1}(S)$$
3. Out : $P^\star$

Transfer scheme

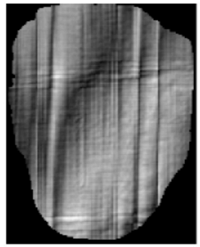
1. Computing $P^\star_{u,cl}$ for all users & classes
2. Using tangent space features :

$$x \Rightarrow \left[\phi_{P^\star_{u,cl}}(\Sigma)\right] \in \mathbb{R}^{S' \times S' \times U \times C}$$

3. LASSO linear classifier in the new space

Transfer scheme

Face



Scrambled Face

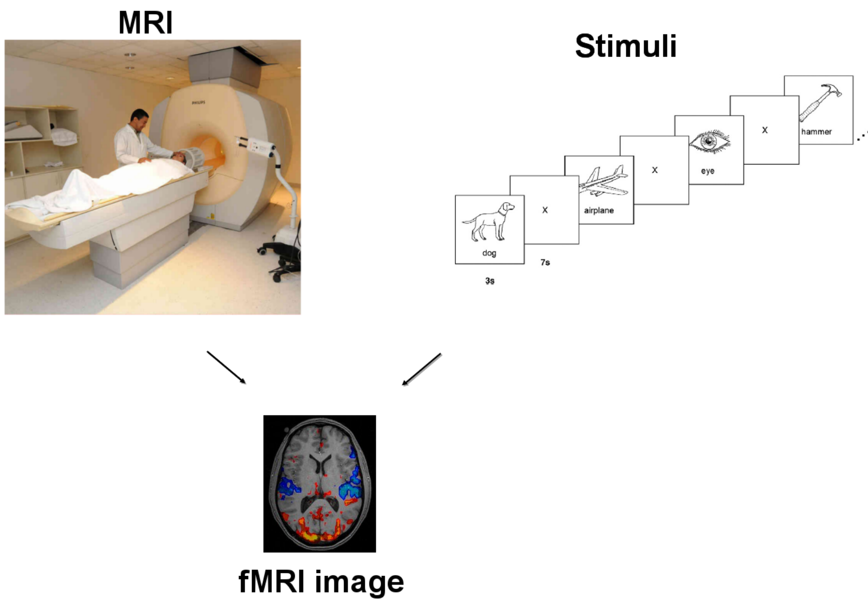
Single trial : 1 sec

#	Δrank	Team Name * in the money	Score
1	—	Alexandre Barachant *	0.75501
2	—	Heikki Huttunen *	0.72668
3	↑21	Nathan Hammes *	0.71316
4	—	hyperplane	0.70227
5	↑1	nagadomi	0.69006

[Kaggle MEG 2014]

Brain Reading : A fascinating task

- Aim : predict the visual stimulus knowing the fMRI
- (the reversed problem is also tackled)

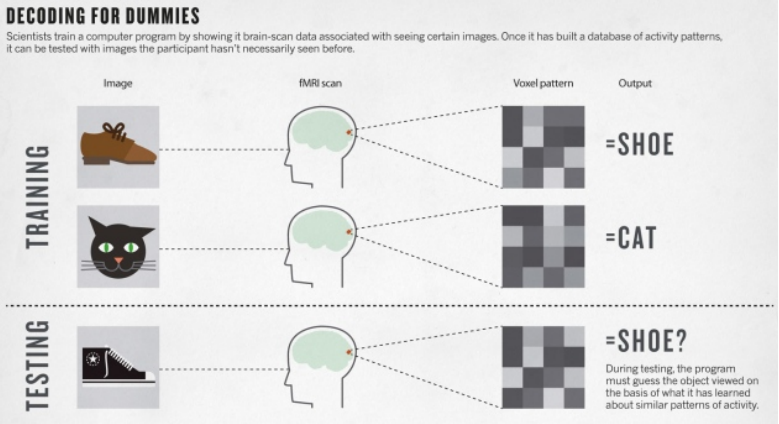


Mitchell et al., Sciences 2008

Predicting Human Brain Activity Associated with the Meanings of Nouns

Brain Reading : A fascinating task

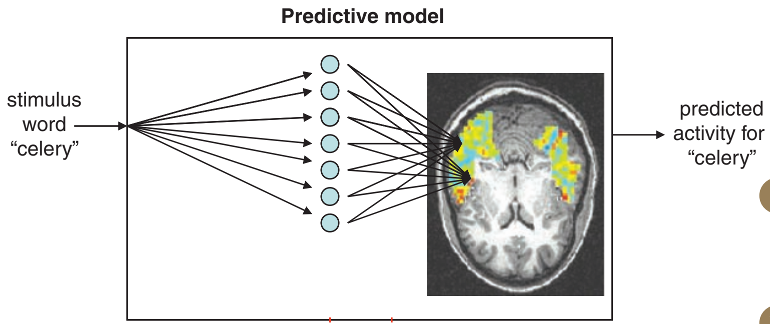
- Aim : predict the visual stimulus knowing the fMRI
- (the reversed problem is also tackled)



Mitchell et al., Sciences 2008
Predicting Human Brain Activity Associated with the Meanings of Nouns

Palatucci et al., NIPS 2009
Zero-Shot Learning with Semantic Output Codes

Predicting brain activity



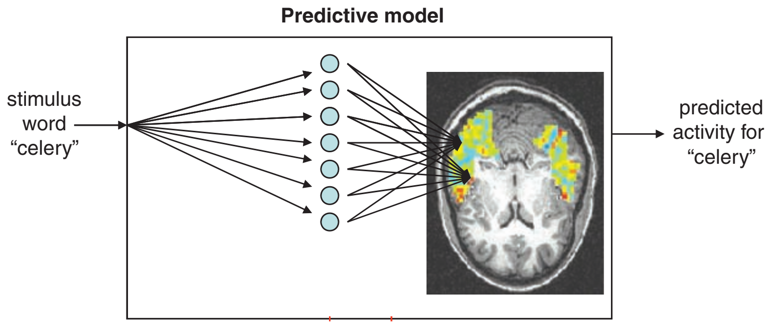
1. How to represent stimuli ? Transformation ϕ
2. How to map ϕ to the fMRI voxel activations ?

$$\text{word } w \Rightarrow \phi(w) \in \mathbb{R}^Z$$

$$\tilde{\mathbf{y}} \in \mathbb{R}^V = \phi(w)R, \qquad R \in \mathbb{R}^{Z \times V}$$

Mitchell et al., Sciences 2008
Predicting Human Brain Activity Associated with the Meanings of Nouns

Predicting brain activity



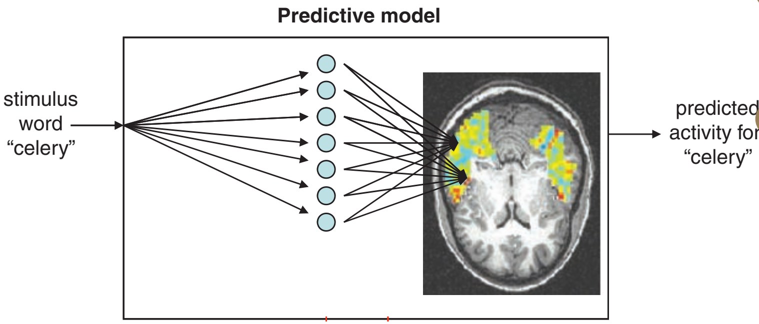
word $w \Rightarrow \phi(w) \in \mathbb{R}^Z$

$$\tilde{\mathbf{y}} \in \mathbb{R}^V = \phi(w)R, \qquad R \in \mathbb{R}^{Z \times V}$$

1. How to represent stimuli ? Transformation ϕ
   - Corpus clustering (ML)
   - Meaningful decomposition (handmade) : "see," "hear," "listen," "taste," "smell," "eat," "touch," "rub," "lift," "manipulate," "run," "push," "fill," "move," "ride," "say," "fear," "open," "approach," "near," "enter," "drive," "wear," "break," and "clean."
2. How to map ϕ to the fMRI voxel activations ?

Mitchell et al., Sciences 2008
Predicting Human Brain Activity Associated with the Meanings of Nouns

Predicting brain activity



1. How to represent stimuli ? Transformation ϕ
2. How to map ϕ to the fMRI voxel activations ? Multi-dimensional regression :

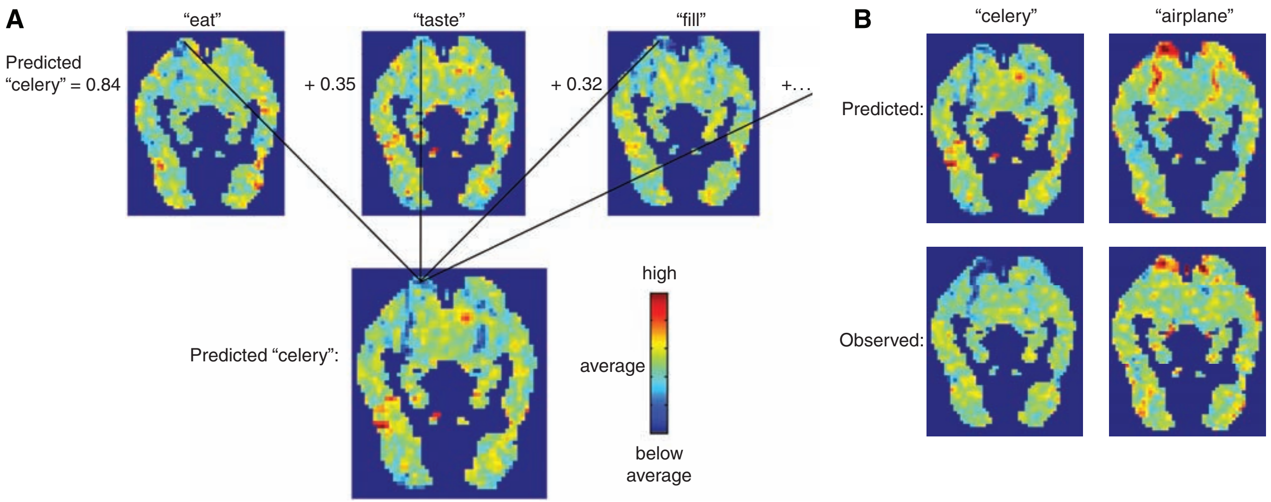
$$\text{word } w \Rightarrow \phi(w) \in \mathbb{R}^Z$$

$$\tilde{\mathbf{y}} \in \mathbb{R}^V = \phi(w)R, \qquad R \in \mathbb{R}^{Z\times V}$$

$$R^\star \in \mathbb{R}^{Z\times V} = \arg\min_R \sum_i (\phi(w)R - \mathbf{y})^2$$

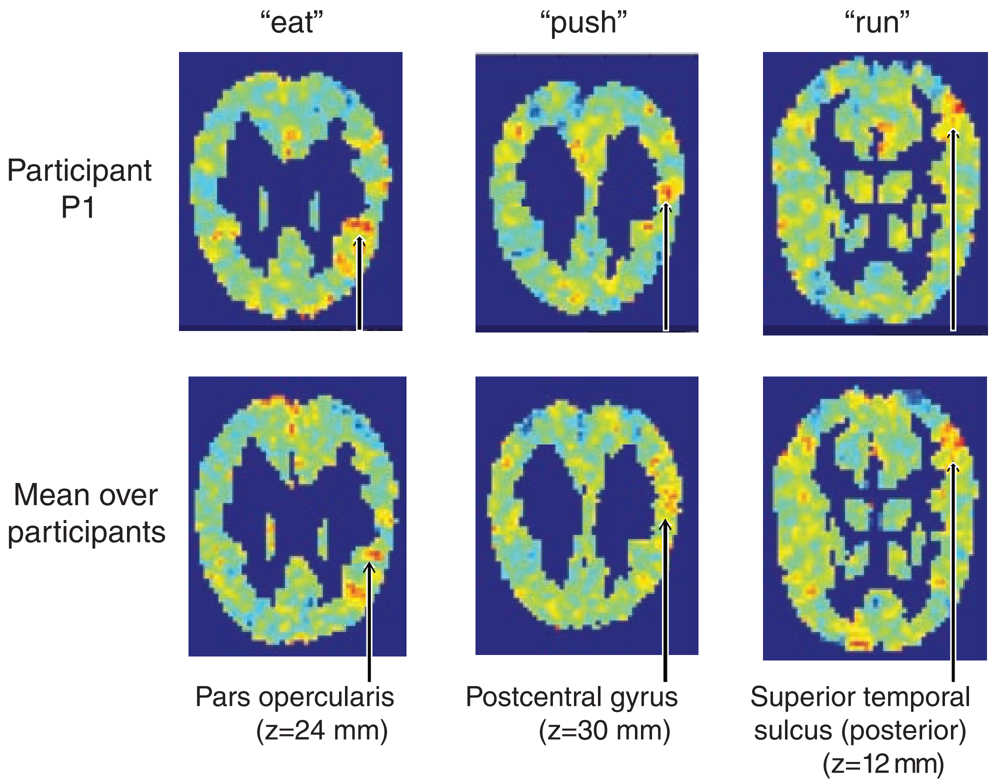
Mitchell et al., Sciences 2008
Predicting Human Brain Activity Associated with the Meanings of Nouns

Interpretation of the results



Linear combination of basic elements

Interpretation of the results

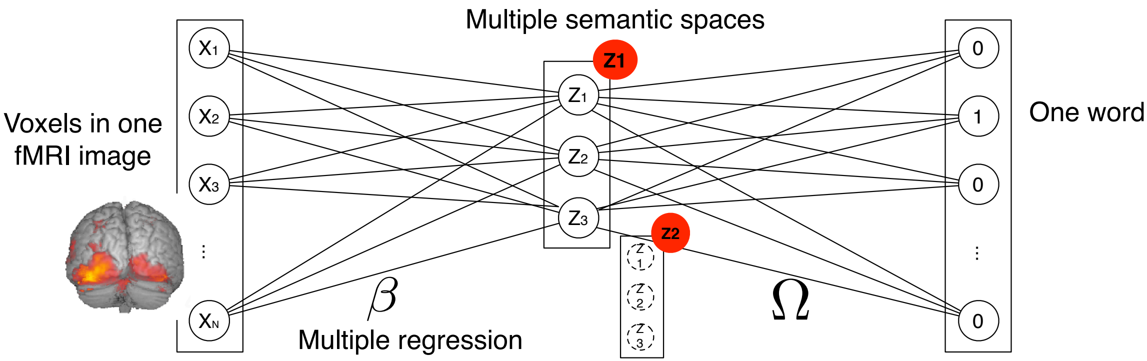


Common points between participants

0-shot learning

An original framwork :

Are we able to find a label that we didn't see in the training step ?

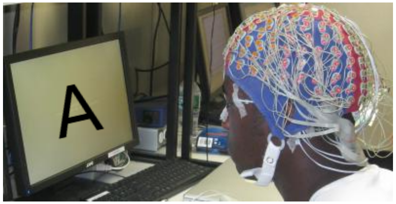


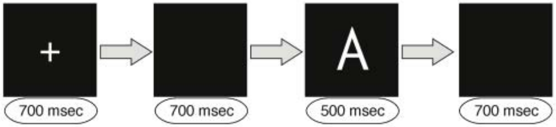
- Ω is a semantic (learned or manually designed)
- Corpus = 60 words ; 58 for training, 2 for testing
- $>$ 80% accuracy (several semantics & bloc regularization)

Pipanmaekaporn et al., 2015
Designing Semantic Feature Spaces for Brain-Reading

Can we tackle brain-reading in EEG ?

- J. Grainger (Marseille) built some datasets

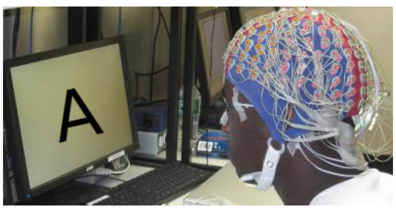


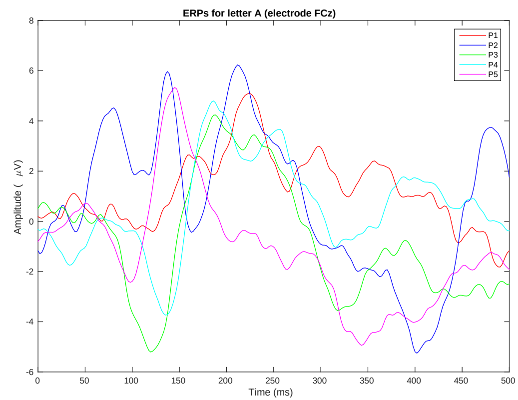


Can we tackle brain-reading in EEG ?

- J. Grainger (Marseille) built some datasets

High variabiliy :



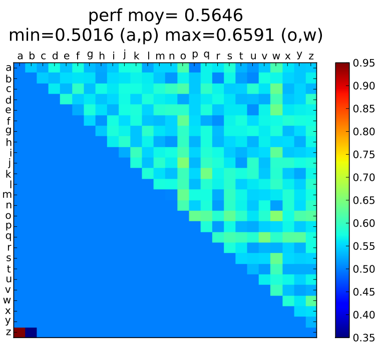


Preliminary results

- SVM $\approx$ Ridge
- Binary classification of couples of letters
  - 325 experiments
  - Baseline (random) = 50%

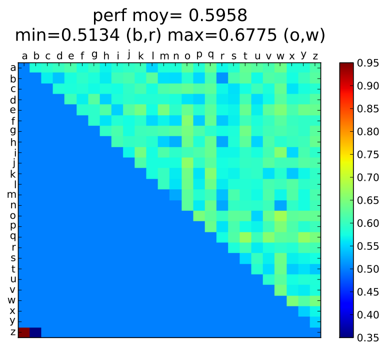
Ridge Regression

56.46%



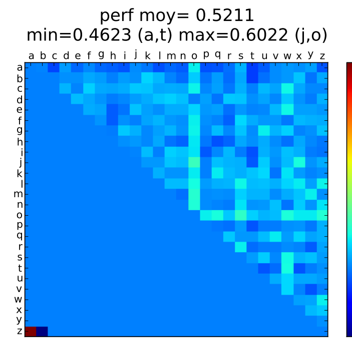
Global

59.58%



Per participant

52.11%



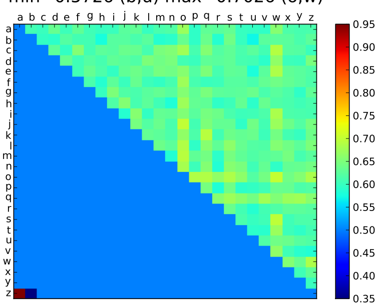
Transfer

Preliminary results

- SVM $\approx$ Ridge
- Binary classification of couples of letters
  - 325 experiments
  - Baseline (random) = 50%

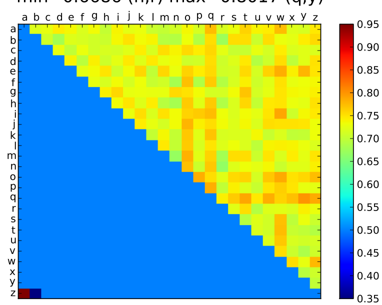
BE-C + Ridge Regression

62.71%



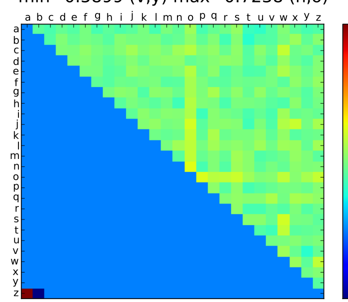
Global

73.28%



Per participant

64.72%



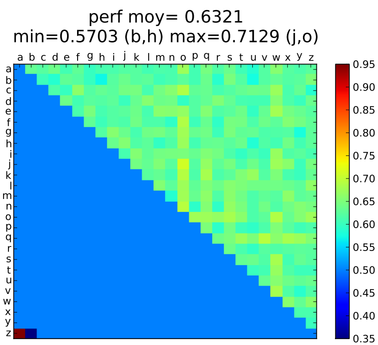
Transfer

Preliminary results

- SVM $\approx$ Ridge
- Binary classification of couples of letters
  - 325 experiments
  - Baseline (random) $= 50\%$

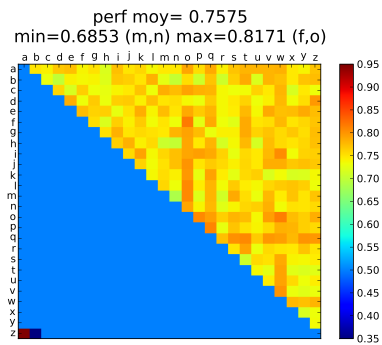
Lasso + Ridge Regression

63.21%



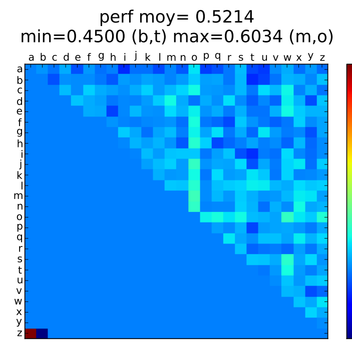
Global

75.75%



Per participant

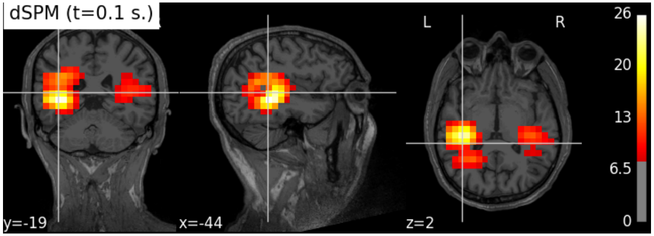
52.14%



Transfer

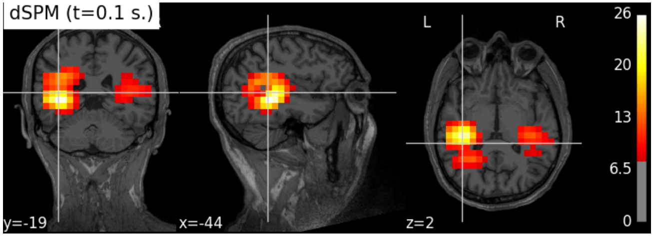
Source Localization

Credit : A. Gramfort



Source Localization

Credit : A. Gramfort



- **X** sources activations in $\mathbb{R}^{P\times T}$, we measure $\mathbf{M} \in \mathbb{R}^{S\times T}$
- **G** gain, estimated by modeling scalp electromagnetic properties s.t. : $\mathbf{M} = \mathbf{GX}$

Inverse problem :

Finding **X** from **M** measurements.

Gramfort et al. 2003
Time-Frequency Mixed-Norm Estimates : Sparse M/EEG imaging with non-stationary source activations

Source Localization

Credit : A. Gramfort

Formulation :

$$\tilde{\mathbf{X}}^{star} = \arg\min_{\tilde{\mathbf{X}}} \|\mathbf{M} - \mathbf{G}\tilde{\mathbf{X}}\|_F$$

Major problem : noise level (!)

A. Gramfort's proposals :

- Using a time-frequency representation
- Exploiting mix-norm regularizations

Conclusion

- Many beautiful problem (from both real life & ML point of view)
- Many existing dataset (BCI, fMRI...)
- Many existing tools (sklearn, mne...)

Let's decode the brain !