A Riemannian Framework for the Classification of Mental Tasks on a Brain Computer Interface

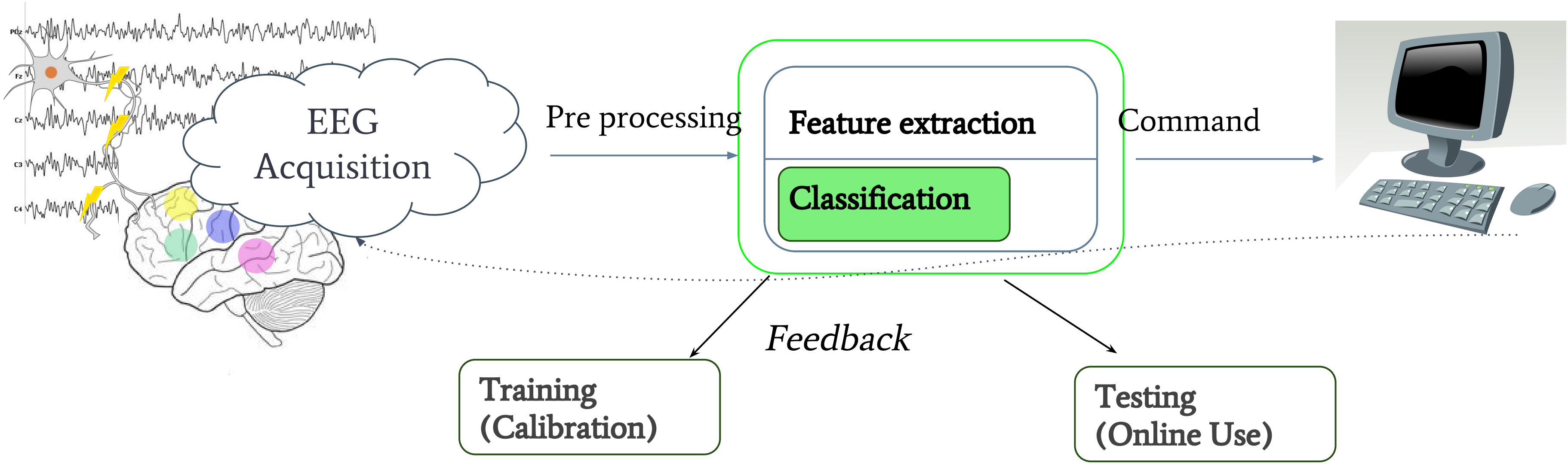
Nathalie Gayraud
Supervisor: Maureen Clerc

BCI-Lift Project/Team
ATHENA Team, Inria Sophia Antipolis - Mediterranee

Introduction: Brain Computer Interfaces (BCI)

“A brain–computer interface is a communication system that does not depend on the brain’s normal output pathways of peripheral nerves and muscles.”

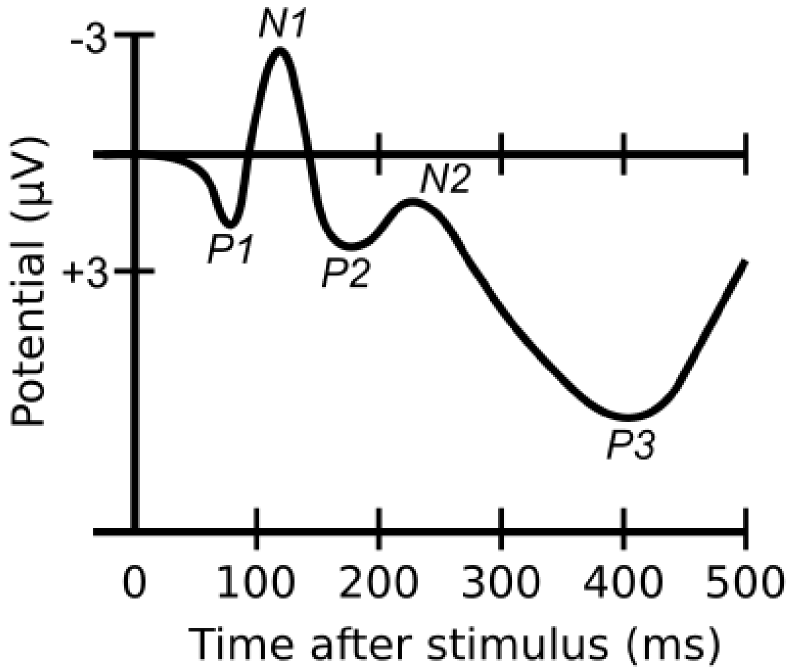
- *definition by Wolpaw et al. [1], 2003*



[1] Wolpaw, Jonathan R., et al. "Brain-computer interface technology: a review of the first international meeting." IEEE transactions on rehabilitation engineering 8.2 (2000): 164-173.

Event-Related Potentials

- Event Related Potentials are brain responses elicited from a **stimulus**
  - Auditory
  - Sensory
  - **Visual**



The P300 Speller

- The P300 wave:
  few “target” stimuli among a train of “nontarget” stimuli.
- A non-invasive EEG-based Brain Computer Interface.
- Spell a letter by counting number of flashes.

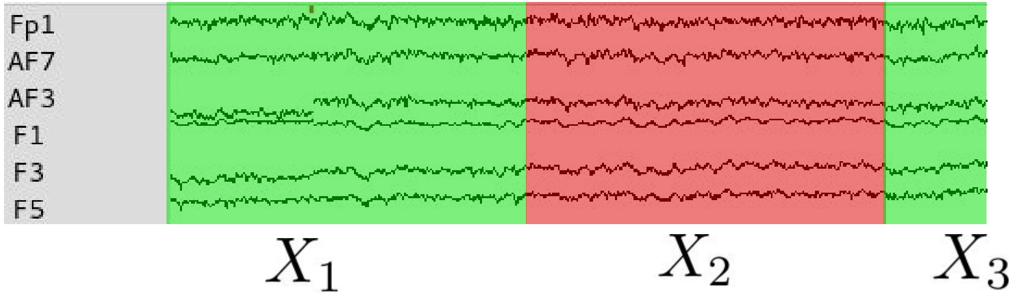
Coadapt P300 speller, calibration.
By Maureen Clerc, Dieter Devlaminck & Loïc Mahé @ Inria Sophia Antipolis-Méditerranée.
Code developed by Inria & Inserm, funded by CoAdapt project (ANR-09-EMER-002).

Pre processing

- Noise filtering
  - Bandpass filtering to eliminate uninformative frequencies.
- EEG segmentation into **I** trials from stimulus onset

$$X_i \in \mathbb{M}(C, N)$$

  - **C** denotes the number of sensors, **N** denotes the time points.

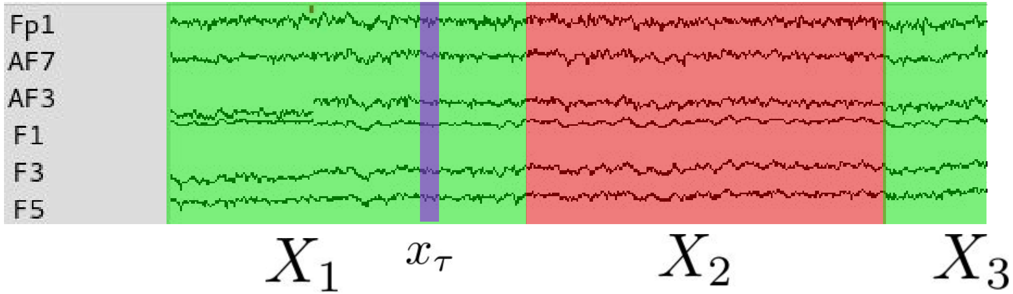


Assumptions

- Choice of N: each trial is a stationary process:

$$x_\tau \sim \mathcal{N}(0, \Sigma^i)$$

  - Trials corresponding to the *target* class follow distribution $x_\tau \sim \mathcal{N}(0, \Sigma^1)$
  - Trials corresponding to the *nontarget* class follow distribution $x_\tau \sim \mathcal{N}(0, \Sigma^2)$



We assume that $\Sigma^1 \neq \Sigma^2$

Feature Extraction

- The features are the elements of the Sample Covariance Matrix

$$\Sigma_i = \frac{1}{N-1} X_i X_i^T$$

- *Assumption:* Σ_i is a Symmetric Positive Definite (SPD) matrix
  - Σ_i lies on the **Statistical Manifold**, also called the **manifold of SPD matrices**.

The Riemannian Manifold of SPD Matrices

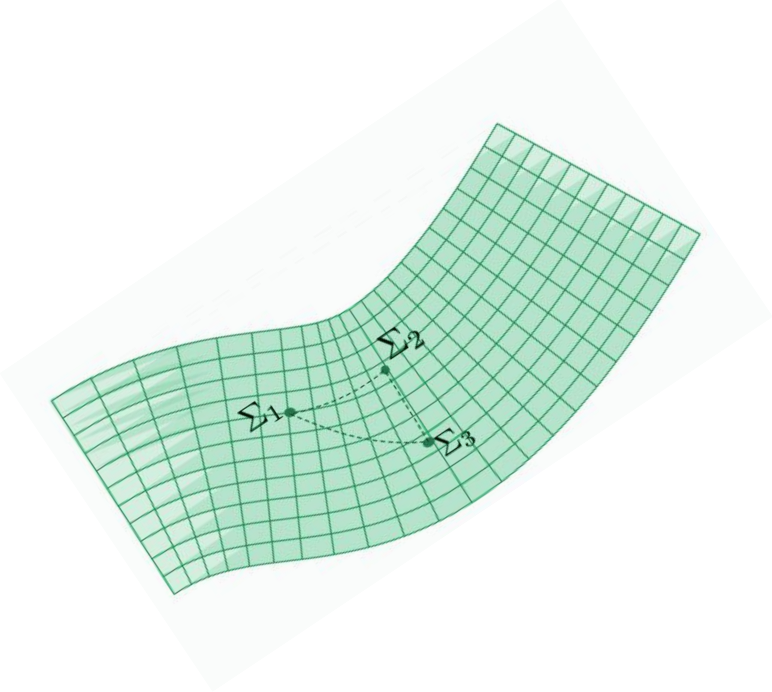
- Embedded with a Riemannian metric

$$d_R(\Sigma_1, \Sigma_2) = \|log(\Sigma_1^{-1}\Sigma_2)\|_F = \sqrt{\sum_{i=1}^{n} log^2 \lambda_i}$$

  - λ_i are the eigenvalues of $\Sigma_1^{-1}\Sigma_2$

- Some properties
  - Hadamard Manifold [2]
  - Derived from information geometry [3].
  - Invariant to linear transformations.

$$d_R(\Sigma_1, \Sigma_2) = d_R(W\Sigma_1 W^T, W\Sigma_2 W^T)$$



[2] Pennec, Xavier. "Statistical computing on manifolds: from Riemannian geometry to computational anatomy." Emerging Trends in Visual Computing. Springer Berlin Heidelberg, 2009. 347-386.

[3] Skovgaard, Lene Theil. "A Riemannian geometry of the multivariate normal model." Scandinavian Journal of Statistics (1984): 211-223.

Materials

EEG signals recorded during P300 speller sessions.
20 Subjects, 3 Sessions per subject (calibration).

C = 12 electrodes
Sampling rate = 256, epoch = 0,5s
N = 128

Bandpass butterworth filter applied.
(5th order, between 1.0 and 2.0)

Two classes:
Target (T), Nontarget (N)

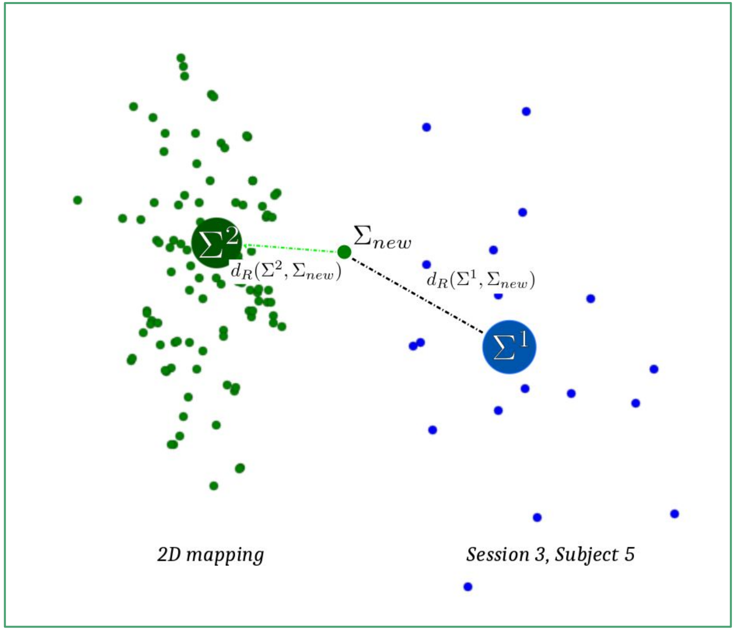
The Minimum Distance to Riemannian Mean Algorithm [4]

- Calibration
  - Compute the Riemannian Center of Mass of each class [5]

$$\bar{\Sigma} = \underset{\Sigma}{\operatorname{argmin}}\left(\frac{1}{I} \sum_{i=1}^{I} d_R^2(\Sigma, \Sigma_i) \right)$$

- Online Use
  - Compute the Riemannian distance to each center of mass
  - The minimum distance defines the classification result

$$class = \underset{c}{\operatorname{argmin}}\ d_R(\Sigma^c, \Sigma_{new})$$



[4] Barachant, A, et al. "Riemannian geometry applied to BCI classification." International Conference on Latent Variable Analysis and Signal Separation. Springer Berlin Heidelberg, 2010.

[5] Fréchet, M. "Les éléments aléatoires de nature quelconque dans un espace distancié." Annales de l'institut Henri Poincaré. Vol. 10. No. 4. 1948.

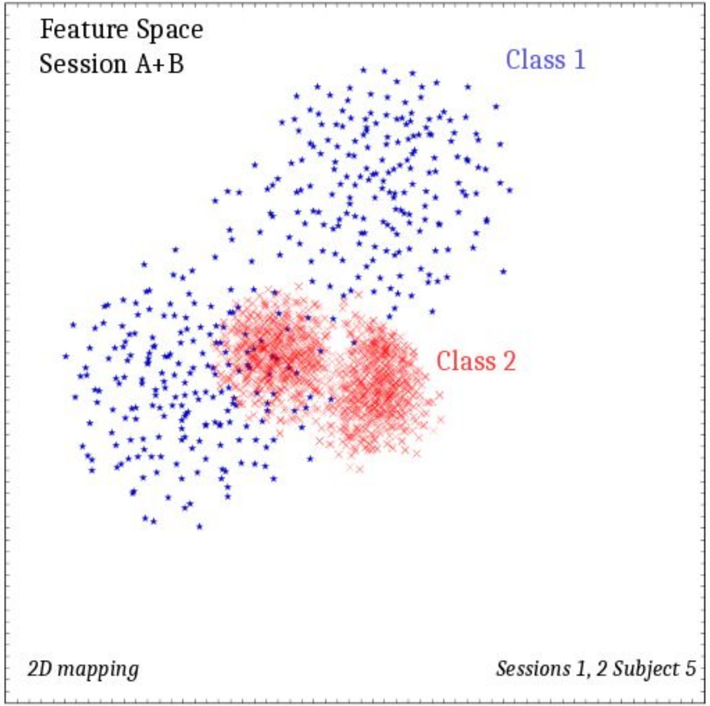
The Problem: Calibrating a P300 Speller

Session 1 - Day 1:

The user is asked to spell a specific word

Session 2 - Day 2:

The user is asked to spell the same word



A solution under the Riemannian Framework

Tangent Space Projection

Tangent space projection of Σ_i at $\bar{\Sigma}$:

$$S_i = \Sigma^{1/2} log(\Sigma^{-1/2} \Sigma_i \Sigma^{-1/2}) \Sigma^{1/2}$$

Projecting S_i back to the manifold:

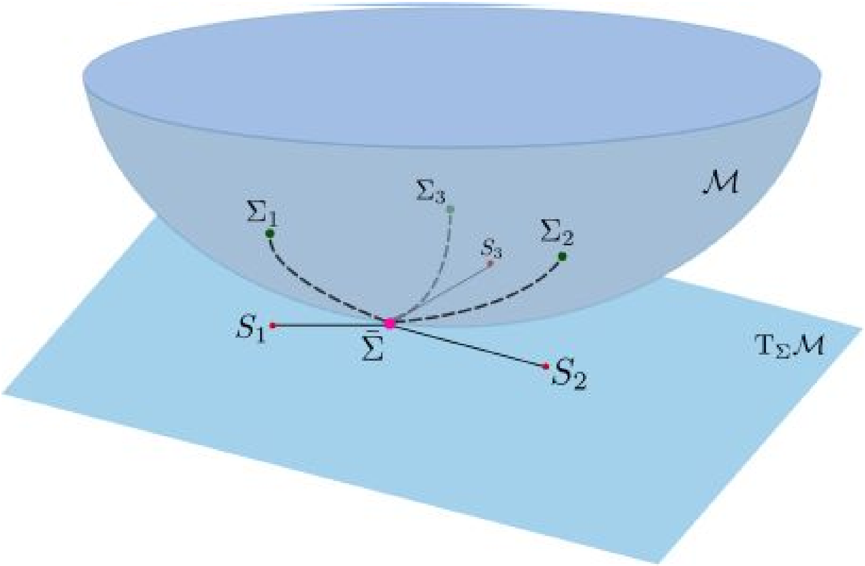
$$\Sigma_i = \Sigma^{1/2} exp(\Sigma^{-1/2} S_i \Sigma^{-1/2}) \Sigma^{1/2}$$

Project the three points on the tangent space at $\bar{\Sigma}$.

Let Σ_1, Σ_2 and Σ_3 be three points on the manifold ***M***

On the Tangent space $T_{\bar{\Sigma}}M$, these points are S_1, S_2, S_3

Compute their riemannian mean $\bar{\Sigma}$



[6] Pennec, X., Fillard, P., and Ayache, N. "A Riemannian framework for tensor computing." International Journal of Computer Vision 66.1 (2006): 41-66.

A Solution Under the Riemannian Framework

- Feature extraction

  Transform the feature space into a ***Euclidean*** *space.*

  - Compute the Riemannian mean $\boldsymbol{\Sigma}$
    - *Where?* Center of Mass of all the Features
  - Project the features onto the **Tangent Space** of the manifold at $\boldsymbol{\Sigma}$
- Train the appropriate classifier

- *Current methods improve classification results*
- Dimensionality Reduction on the manifold
- Robust features, less sensitivity to outliers
- Classification Algorithms using Differential Geometry

Discussion

Thank you for your attention!

Questions?