Decoding Brain Signals

Microsoft machine learning competition

Ruslan Aydarkhanov, 4th place

August 20, 2016

Background

Faculty of bioengineering and bioinformatics

MSc in neurosciences

Minor in computational neurosciences

Doctoral program in neuroscience

Brain-Computer Interfaces

Basic machine learning courses during Masters + self-education

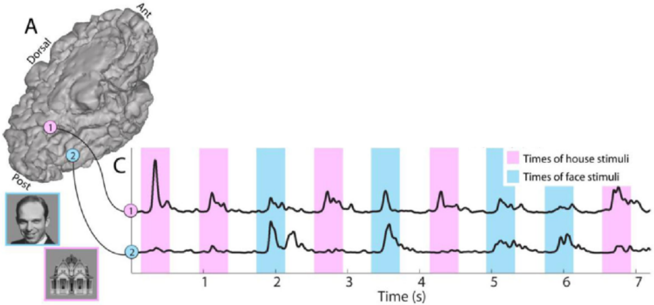
Azure ML platform

No access to testing dataset
Solutions are submitted as Azure ML services and tested remotely
No teams

1. Pure Azure ML
   - Fast development of simple models
   - GUI, drag and drop
   - Not flexible and limited in tools
2. Building models locally
   - Python + all its package zoo
   - Same structure for all projects on Azure ML
   - Outdated scikit-learn v0.15.1 on Azure ML (newest is v0.17.1) -> virtualenv
   - Learning and validation on a laptop

Competition

Decoding Brain Signals



Miller et al, 2014

- Electrocorticogram (ECoG)
- up to 64 channels
- window $\pm$ 400 ms around stimulus presentation onset
- 4 patients
- 200 training samples per patient
  40 samples in public dataset
  60 samples in private dataset
- Performance metric: classification accuracy

Data processing

Basics of EEG/ECoG processing

- Spatial filtering and decomposition
  - ✓ common average reference (CAR)
  - laplacian
  - PCA, ICA, CSP …
- ✓ Band-pass filtering in time
- Frequency band power
  - Power spectral density
  - Signal envelope (hilbert transform)
  - ✓ Wavelet transform

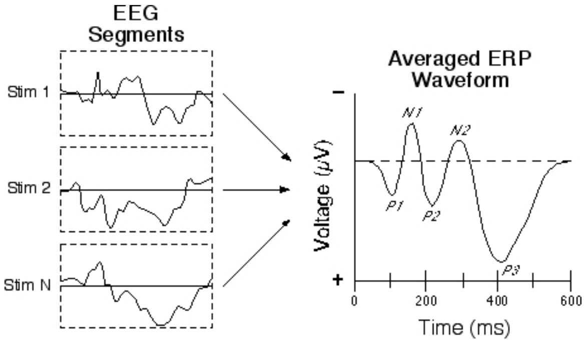
Features

- ✓ Event-related potential (ERP)
- ✓ Wavelet transform
- ✓ Event-related broadband (ERBB)
- ✓ Covariance matrix projected on a Riemannian tangent space

Feature selection

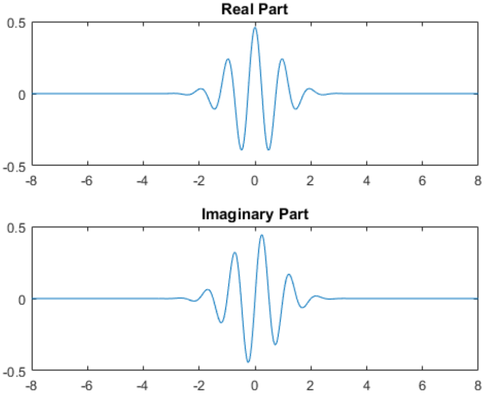
- ✓ ANOVA
- Genetic algorithm

Event-related potential (ERP)



- Bandpass filtering 1-10 Hz
- Window from -50 to 350 ms
- Down-sample by factor of 30

Wavelet transform



Continuous wavelet transform

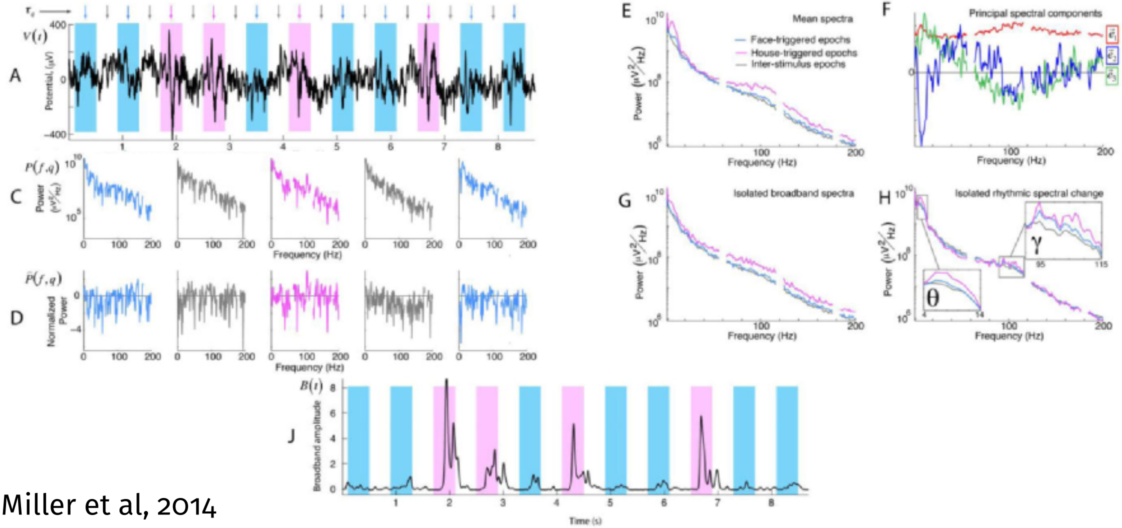
- Complex morlet wavelets for 4-10 Hz

  $$e^{j*w*x} * e^{-x^2/2}$$

- Convolution + absolute value -> Instant frequency power estimation
- Window from -50 to 350 ms
- Down-sample by factor of 30

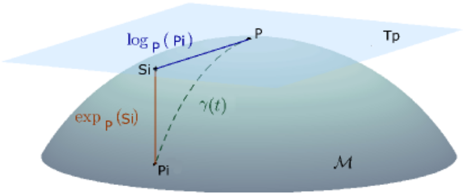
Python library for EEG: MNE

Event-related broadband (ERBB)



Miller et al, 2014

Riemannian projection



- Compute covariance matrices with shrinkage from raw data
- Pick up the projection reference point
- Project on a tangent space

Python library: pyRiemann

Developed by the winner Alexandre Barachant

Originally for neural data, but can be applied to any set of semi-positive definite (SPD) matrices, e.g. covariance matrices

Classifiers

Simple classifiers:

- Arbitrary combination of features
- Individual classifiers for patients
- Logistic regression with patient-specific L1 penalty (grid search)
- Stratified 5-fold cross-validation

Ensemble:

- Weighted average of 8 best first-layer classifiers
- Stacked classifier
  - Input is 5-fold out-of-sample predictions
  - SVM with CV Grid Search of parameters
  - No validation

Intermediate progress

- Local cross-validation correlated well with leaderboard
- Leader position for 1 month with one ERP model (score 81)
- Adding more features, learning multiple models
- Averaging and stacking
- Small testing set (4 x 40) -> struggle for each sample in the end
- Both private and public scores are very noisy

Results

Name	Highest Public Score	Highest Private Score	Final Ranking
Alexandre Barachant	**90.625**	93.75	1
KyuHwa Lee	**88.75**	92.5	2
Jean-Remi King	86.25	88.3333	3
Ruslan Aydarkhanov	**89.375**	88.3333	4
Marouane FELJA	82.5	84.1667	5
Igor Inozemtsev	81.875	84.1667	6
lmzintgraf	**89.375**	82.9167	7
Pablo Seibelt	81.25	82.5	8
Carlos Aranda Torres	81.875	82.5	9
Bruce Cragin	77.5	82.0833	10

My solution: https://github.com/Aydarkhan/decoding_brain_signals_MS

Leaders' solutions

- Alexandre Barachant, 1st place
  - Average of 5 models of L2-penalized logistic regression
  - ERP, Riemannian projection
  - **Covariance matrices in frequency domain** projected on tanget space, averaging models across frequency bins
  - **Cross-covariance matrices** (covariance of Hankel matrices)
- Kyuhwa Lee, 2nd place
  - Average of 2 models of **Gradient Boosting Machine**
  - PSD with multitaper on 2 windows 0-300ms and 100-400ms
  - Instant band power with Hilbert transform in low band < 10 Hz and high band 10-70 Hz
- Jean-Remi King, 3rd place
  - **Bagging** of 5 models of logistic regression
  - ERP, Riemannian projection
  - PSD and time-frequency features

Lessons and take home messages

- Competitive spirit boosts your efficiency
- Tried many things in short time

- Write good code from the beginning
- Feature engineering is crucial -> check the literature (especially with small and high-dimensional datasets)
- For brain data take into account
  - time - channel
  - spectral characteristics
  - interaction between channels
- Ensemble methods are powerful

Thank you for your attention!

Questions