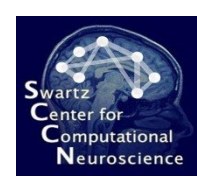




Exercise 1: Implementing ERP-based BCIs

Introduction to Modern Brain-Computer Interface Design

Christian A. Kothe
SCCN, UCSD



About

- This is a programming exercise in pure MATLAB (no toolboxes!)
- You will be implementing the critical parts of an ERP-based BCI: *shrinkage LDA*, the *prediction function*, and the *loss function* that evaluates the mis-classification rate
- **Tip 1:** You may want to take a look at the previous lecture slides to find the equations for what you will be implementing
- **Tip 2:** Be prepared to take a look at your intermediate results in the command line to fix any issues with wrong matrix shapes and so on

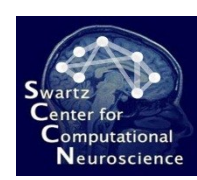


Preparation

- Start MATLAB
- To get into the directory, type:
`cd /your/path/to/exercise_package/ex1`
- To open the evaluation script (that runs your code) type:
`edit run_evaluation`

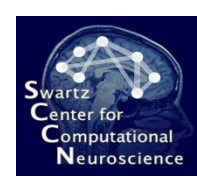
Details

- In this folder you find the scripts for this exercise
- **run_evaluation.m**: This function loads the data and runs your BCI functions
- **train_erp.m**: This is an incomplete scaffolding of the calibration function for this approach, here you need to fill in the code for the shrinkage LDA
- **test_erp.m**: An incomplete scaffolding of the prediction function: you need to fill in the LDA classification
- **eval_mcr.m**: An small stub function to compute the offline mis-classification rate on the test set: you need to fill in the loss calculation



Exercise

- Implement the missing parts so that the `run_evaluation` function completes successfully
- The final output is the percentage of misclassified trials in the data, it should be about 8% if your code is correct



For MATLAB Pros

- There are a few things that can be improved in the code, among others:
 - Implement a band-pass filter using fft/ift before you extract features (replaces the mean subtraction)
 - Try to play back the data through your prediction function in a loop and plot the results
 - Implement the prediction function so that it can take chunks of any length and aggregates them into a sliding window/buffer across calls



For MATLAB Newbies

- Mean: mean
- Covariance: cov
- The “s” variable in shrinkage LDA can be set to 1
- Bias should be a number (not a matrix)
- The trialfeatures matrix in test_erp.m can be turned into a vector by writing:
trialfeatures(:) – this is a single trial
- Make sure that your test_erp outputs either -1 or +1

