Common spatial pattern (CSP) sensorimotor rhythm (SMR) BCI

The aim of this exercise is to implement the basic CSP algorithm, to classify CSP-features with an LDA classifier and to validate the performance. Further, we compare the CSP results to results computed by using a Laplacian spatial filter.

Exercise 6.1: CSP Algorithm

Implement the conventional CSP algorithm following the steps below:

- Training data: `Sig.mat`. First dimension 15 mono polar channels, second dimension time samples. The time/sample information on when cues were presented to the subject are stored in `ClassPosH.mat` for hand class and `ClassPosF.mat` for feet class. The sample rate in `SampleRate.mat`.
- Band pass each channel of your training data between 7 and 30 Hz. Use Matlab function `butter` for the filter design. 4th order is enough. Use Matlab function `filtfilt`.
- Prepare a data matrix `X`. To this end, epoch the band passed channels according to the class positions. For each trial, extract the segment from 1.5 to 2.5 seconds after the cue presentation. Each sliced window should consist of m channels x n time samples. After successful segmentation, you have a matrix `X` of dimension 100 trials x 15 channels x 128 time samples. Do not forget to construct a vector for your class labels `Y` (dimension 100 trials x 1 class label).
- Calculate the separate spatial normalized covariance Σ for each epoch according to

$$\Sigma = \frac{XX^T}{trace(XX^T)}$$

- Average covariance matrices for each class separately using:

$$\Sigma_1 = \frac{1}{N_1}\sum_{i=1}^{N_1}\Sigma_i \text{ for class 1 and } \Sigma_2 = \frac{1}{N_2}\sum_{i=1}^{N_2}\Sigma_i \text{ for class 2.}$$

- Solve the eigenvalue problem with the Matlab function `eig` by using the ‘`qz`‘ algorithm. The ‘`qz`′ algorithm is more stable but slower than the standard algorithm:

$$[V, D] = eig(\Sigma_1, \Sigma_1 + \Sigma_2, 'qz')$$

- V is our transformation matrix (CSP filter) of dimension 15x15. Each column is a spatial filter. Sort the columns according to the eigenvalue magnitude. After sorting, first and last column are the most important transformed channels.

Exercise 6.2: Extract CSP features and train an LDA

We want to combine the CSP filter with an LDA classifier.

- Compose a filter matrix `C` (dimension 4 filters x 15 filter weights) out of the V matrix. Use the two most important filters for each class (Column 1 to 2 and 14 to 15).
- Multiply the band passed signal from exercise 6.1 (denoted as SigBand) with the CSP filter Matrix.

$$SigCSP = C^T \cdot SigBand$$

- After applying the CSP filter the SigCSP is of dimension 4 channels x (many) time samples. These 4 "virtual" channels are linear combinations of the original 15 channels.
- Epoch CSP filtered channels as described in exercise 6.1. Do not forget to store the class labels of each epoch (`Ytrain`, dimension 100 trials x 1 class label).
- Calculate the `log10` band power of each CSP channel and each epoch separately. Store the log band power values as your training data for the LDA classifier (`Xtrain`, dimension 100 trials x 4 log band power features).
- Train an LDA classifier with `Xtrain` and `Ytrain`.

Exercise 3: Validating the CSP+LDA BCI on unseen data

Validate the CSP+LDA BCI on unseen data with different amounts of columns from V matrix.
Outcome: Classification accuracies in dependence of the amount of columns of the V matrix.

- Load the `.mat` data with the appendix `Val`. The content and the naming is the same as in exercise 6.1, except the amount of data is less with only 60 trials (30 per class).
- Band pass the validation data (7 to 30 Hz).
- Apply the CSP filter to the band passed validation data.
- Perform segmentation (epoch) and log band power calculation of the segments (epoch), like in exercise 6.2.
- Use the obtained features for assembling your validation matrix (`Xval`, 60 trials x log band power features). Construct an `Yval` vector containing the class labels.
- Classify your `Xval` data with the LDA classifier. Calculate the classification accuracy.

Repeat the classification with different amounts of columns of the V matrix. Start with using the only the first and the last column (leading to 2 virtual channels) and end with using the first 7 and the last seven columns of the V matrix as the CSP filter matrix. What is the course of the accuracy over the amount of V matrix columns? How can you explain the shape of the course?

Exercise 4: Discuss your results

The accuracy achieved with Laplacian derivations as spatial filters is approximately 78%. What are your CSP results? Are there differences in performance? When so, then please explain what you think is causing the differences!

Pre-submission via TeachCenter at the end of the unit; final submission 23.06.2018