P300 Evoked Potential (EP) feature extraction

Exercise 2.1: P300 Evoked Potential

In unit #1 we computed target and non-target P300 curves by averaging individual EEG segments (trials). Fetch the code and let's continue with the P300 analysis…

Exercise 2.2: Calculate the standard deviation of the mean

Mean EP curves are typically computed by using a limited number of trials. Hence, we cannot be sure that we obtain the "real" EP curves. To get a better understanding of whether or not individual curves are different, we want to know the typical deviation of the mean (a.k.a. standard deviation (SD)).

Task:

- Compute the standard deviation (SD) of the EP's for every time-point of each condition of exercise 2.1 (the MATLAB function `std` computes the standard deviation).
- Integrate plots of the standard deviation in the plots of exercise 2.1 (`hold on`).

Exercise 2.3: Statistically significantly different?

Now we want to test whether or not target and non-target EP are different. Use the Wilcoxon test (Matlab `ranksum`) to compute a statistical test for each time point of each channel between non-target and target. Wilcoxon is suitable as the amount of trials is different between the conditions and the distribution of the trials is unknown. Use a significant level of $\alpha=0.01$. Do not forget to correct for multiple comparisons (Bonferroni correction, divide significant level by the amount of comparisons).

Task:

- At which time points and which channel are the two modalities (target, non-target) from exercise 2.1 statistically significantly different? Make plots!
- Where is the biggest coherent cluster of differences in each channel?

Linear Discriminant Analysis (LDA) classifier

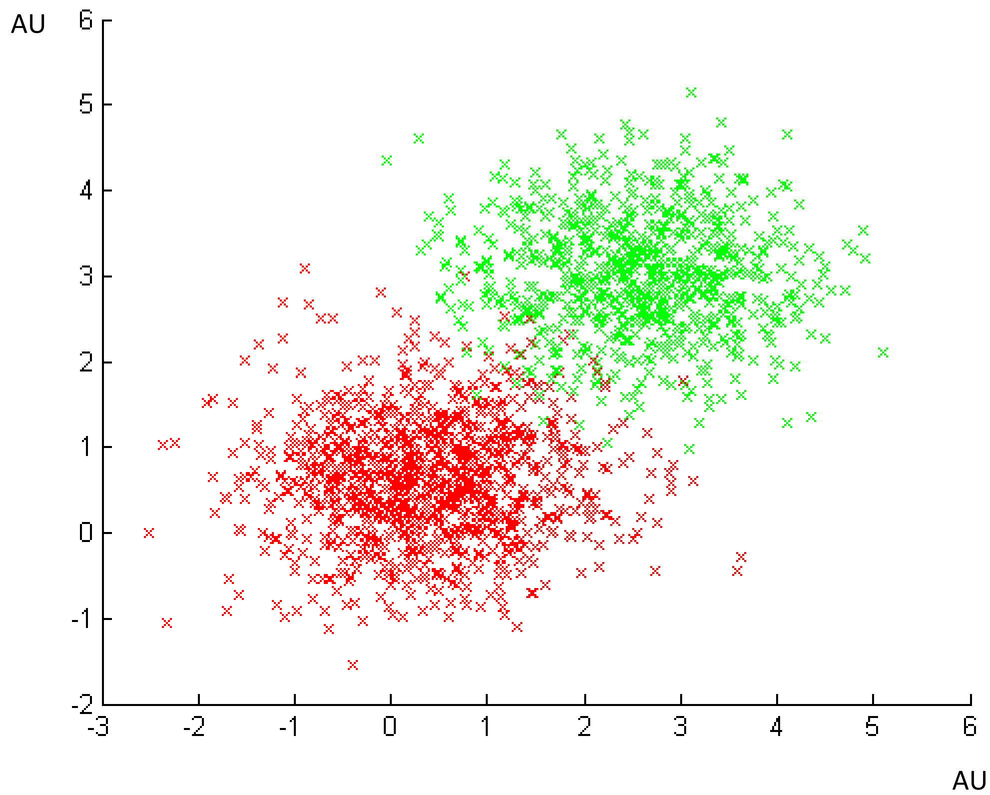
Exercise 2.4: Generate artificial training and testing data

Before implementing the LDA classifier, we want to create artificial training and test data sets. The data will be useful to test and evaluate the LDA implementation. Create a two-dimensional test and training data set for two different classes. Data for the individual classes should be drawn from two different normal distributions (`randn`). Use a center of 0.4/0.6 (dim 1/dim 2) and a standard deviation (SD) of 0.9/0.7 for the first class. For the second class, use a center of 2.5/3.0 (dim 1/ dim 2) and a SD of 0.9/0.7. For LDA training generate 1100 class 1 trials and 800 class 2 trials. To test the LDA generate 500 class 1 and 400 class 2 trials.

Task:

- Create matrices `X_train` and `X_test` that contain the trials in the columns and the dimensions in the rows.
- Create column vectors `Y_train` and `Y_test` that contain the corresponding class labels.
- Visualize the distributions (`scatter`).

Toy example:



Exercise 2.5: Implementation of a custom LDA

The idea behind linear discriminant analysis (LDA) is to use hyperplanes to subdivide the feature space in class-specific subspaces. In the case of a binary decision (2 subspaces), the classification of a given sample x is performed by

$$\mathrm{sign}(\mathrm{w}^{\mathrm{T}}\mathrm{x} - \mathrm{b}), \qquad (1)$$

where b denotes the bias (offset of the hyperplane) and w the projection vector. The projection vector can be calculated by

$$\mathrm{w} = \Sigma_c^{-1}(\mu_1 - \mu_2), \qquad (2)$$

where μ_i denotes the vector of estimated means per feature dimension of class i and Σ_c is the estimated common covariance matrix (the average of the individual class covariance matrices). The bias is given by

$$b = w^T \mu$$

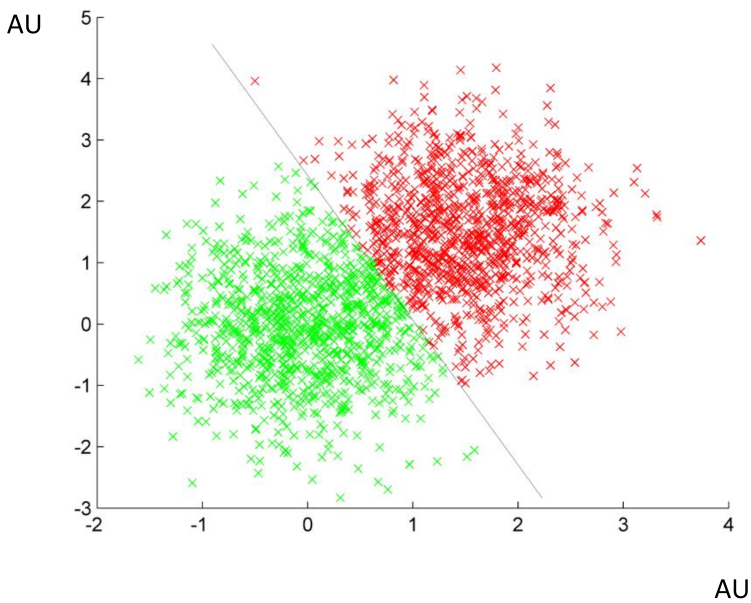
where μ denotes the vector of estimated means per feature dimension over all classes.

Implement an LDA classifier (function name `custom_LDA`) based on the above equations.

Task:

- Use your artificial test data to test the LDA classifier and plot your results (`scatter`)
- Highlight the decision hyper plane (in the two-dimensional case a line).
- What accuracies do you get?

Toy example:



Exercise 2.6: Classification of EP training and testing data

Create training and testing data out of the real AEP data that we used in the last exercise (`BI5_segments_HTS.mat`). Take the time point of channel Cz with the lowest p value (most significant different) at around 300ms for dimension 1 and the most significant time point at around 500ms for dimension 2. Visualize the distributions (`scatter`). Divide the data as following: 1000 non-targets and 200 targets for training; 800 non-targets and 200 targets for testing. Select the trials random subsampling.

Task:

- Create the same data structure as in exercise 1 (`X_train`, `X_test`, `Y_train`, `Y_test`).
- Test if the data are normal distributed to make sure they are suitable for classification with LDA. Use `normplot()` for each of the classes (target, non-target) and dimensions separately.
- Use your test data to test the LDA classifier and plot your results (`scatter`)
- Highlight the decision hyper plane (in the two-dimensional case a line).
- What accuracies do you get?

Pre-submission via TeachCenter at the end of the unit; final submission 07.04.2019