Lecture 7 exercises

AI for Medical Time Series - Spring 2022

Introduction

In this exercise set, you will practise PCA decomposition for time-series data. This week you will be working with an Electrocorticography (ECoG) dataset (described here: https://doi.org/10.1016/j.cub.2018.07.045). This dataset contains ECoG data which are recorded from the orbitofrontal cortex in 10 patients with epilepsy. The data can be found as a 2D numpy array "data_all_buttonpress_window_events_hg.npy" with dimensions time-samples by electrodes. The data file is available on Ilias.

Important: The purpose of this exercise is to let you explore how PCA is implemented. For this we ask you to solve them writing your own code, i.e. without using built-in PCA and SVD functions available in existing python libraries.

The exercise will be marked as OK if you get **19** / **24** points or more.

The solutions must be handed in via **ILIAS**. Deliver your submission as a compressed file (zip) containing one .py or jupyter notebook file. Please make sure to name the zip file as follows:

HW_homeworkNumber_surname_name.zip.

If you are working as a group, then indicate the two names in the file name as

HW_homeworkNumber_surname1_name1_surname2_name2.zip.

Please use comments to indicate which sub-task you are answering within the main exercise (# Exercise 1a, etc.). Questions that do not ask for code can be answered as comments, or as markdown if you are using Jupyter Notebooks.

The exercises can be handed in by two students working together. Copying code or solutions of individuals outside the group (e.g. submitting the code of other individuals as your own) will result in 0 points.

Deadline: 16:00, **April 27**.

Exercises

1. **Principle Component Analysis (PCA)** .... 24 points

   (a) **Implementing PCA** .... 14 points

   Write a function that decomposes the input data into given number of components by implementing the PCA algorithm. This function should take as input a 2-dimensional time domain signal (time samples by observations, which in our case correspond to different electrodes) and the desired number of components.

   Important: You are allowed to use built-in functions for covariance and eigenvalue or eigenvector computations (e.g. numpy.linalg.eig or scipy.linalg.eig). However solutions with existing decomposition methods (SVD, Truncated SVD, PCA, etc.) **will not be accepted.**

   After computing the principal components, which are defining a new feature space as a linear combination of the original feature space (electrodes in our case), transform the data to the new feature space. You can do that by multiplying the data with the set of components. Keep in mind that you may need to normalize the data.

   Compute the percentage of variance that is attributed by each of the selected components.

   Return the transformed data and total percentage of explained variance with the given number of components.

   (b) **Finding the optimal number of components** .... 5 points

   Import the data file "data_all_button_press_window_events_hg.npy" and use this ECoG dataset as an input to your PCA function.

   Experiment with the number of retained principal components. To do that, use different component numbers ranging from 1 to the number of channels. Report what in your opinion is the optimal component number for the ECoG dataset, and describe how you chose this number.

   (c) **Visualizing clustering results** .... 5 points

   Use the optimal number of components that you identified to transform the data. Plot the transformed data.