 Aviv Dotan, Daniel Poliakov, Nir Getter, Erez Wolfson

Exercise #5 – PCA of Epileptic Seizures

The utilization of all the techniques and functions described in this document are mandatory in the submitted code.

Background

- PCA is a popular method for dimensionality reduction and data compression.
- The goal of this exercise is to get familiar with PCA through analyzing a dataset of EEG recordings from an epileptic patient prior to 3 different seizures.
- Through the exercise, you will also get familiar with some common analyses of EEG data.

1. Data description

- The EEG recordings were taken using the following 19 electrodes:
  `Fp1, Fp2, F3, F4, C3, C4, P3, P4, O1, O2, F7, F8, T7, T8, P7, P8, Fz, Cz, Pz`
- Each recording begins 100 minutes before a seizure and ends at the beginning of a seizure.
- The sampling rate is 250Hz.
- The EEG dataset has already gone through some basic EEG processing methods (e.g. bandpass filtering to 1–40Hz and a basic artifacts removal procedure).
- Each recording was saved using the following naming format:
  `'p<# of patient>_s<# of seizure>.mat'`
- Each file contains a single matrix named **`data`**, where each row corresponds to a recording from a different electrode.

2. Loading the data

- You should place the data files in a dedicated folder named `'..\DATA_DIR'`.
- Your code should automatically search for all files that match the data files naming format and ignore all other files and subfolders in the data folder:
  - Use the **`dir`** function to get a list of all files in the data folder.
  - Use the **`regexp`** function to test whether a file name matches the naming format.
- Your code should automatically extract the patient and seizure numbers directly from the file name. Use any necessary functions (e.g. **`regexp`**, **`strfind`**, **`str2double`**).
- Every file found to match the naming convention should be loaded and analyzed.

3. EEG Feature Extraction

In the following section, all given parameters (e.g. frequency bands or time windows) are just a recommendation. You should fiddle with them to make your results look as clear as possible.

- Split the EEG data into time windows. Each window should be of 40 seconds and start 20 seconds after the previous window (there is an overlap between the time windows). Ignore any overflowing time windows (instead of zero-padding them).
- You will extract 342 features from each time window (18 features per electrode). Your code should preallocate the needed memory in advance. Note that the number of features depends on the number of electrodes and frequency bands.
- From each time window, extract the power of the signal.
    - Use Welch's method (**`pwelch`**), to estimate the power spectrum within each time window. We will use the notation $p(f)$ to describe its output.
    - For the Welch's method, use time windows of at least 2 seconds. It is highly recommended to use overlapping time windows.
    - Read the **`pwelch`** documentation and make sure to use it properly.
    - **Do not use `pwelch`'s defaults** – calculate and explicitly send all the parameters to the function.
- For the first couple of features (*relative power* and *relative log power*), split the power into frequency bands (calculate the sum of the power within each band):
    - *delta* 1–4.5Hz
    - *theta* 4.5–8Hz
    - *low alpha* 8–11.5Hz
    - *high alpha* 11.5–15Hz
    - *beta* 15–30Hz
    - *gamma* 30–40Hz

  Make sure your code can handle changes in the frequency bands' definition (i.e., different limits or different number of bands).
  Make sure not to overlap frequency bands.
- For the last three features (*spectral moment*, *spectral edge* and *spectral entropy*) you should normalize the raw power by the total power so it can be treated as a probability function (do not split into frequency bands this time):

$$p_{norm}(f) = \frac{p(f)}{\sum_{\nu} p(\nu)}$$

- Extract the following features from each time window:
    - Compute the *relative power* of each frequency band:
        - Within each electrode, divide the power of each frequency band by the total power:

$$relative\ power\ (band) = \frac{\sum_{f \in band} p(f)}{\sum_f p(f)}$$

- You should get **`n_electrodes*n_freq_bands`** (i.e. $19 \cdot 6 = 114$) new features.

- Compute the *relative log power* of each frequency band:
  - Repeat the computation of the *relative power* using the log-power instead of the power:

$$relative\ log\ power\ (band) = \frac{\sum_{f \in band} \ln(p(f))}{\sum_f \ln(p(f))}$$

  - To make sure there are no nonnegative values of the log-power, replace $\ln(p(f))$ using the following trick:

```
log_power = log(exp(1).*power./min(power));
```

    - In your report, explain how this trick prevents negative values of log-power and the importance of it.
  - You should get **`n_electrodes*n_freq_bands`** (i.e. $19 \cdot 6 = 114$) new features.
- Compute the *root total power*:
  - Calculate the square-root of the total power of each electrode:

$$root\ total\ power = \sqrt{\sum_f p(f)}$$

    This feature is proportional to the standard deviation of the original signal.
  - You should get **`n_electrodes`** (i.e. 19) new features.
- Compute the *spectral slope and intercept*:
  - Draw a plot of $\ln(power)$ of each electrode as a function of $\ln(frequency)$. This plot should be roughly linear (with some deviations, especially in the *alpha* band).
    In your submission, the code should NOT display this plot; it is only meant to help your understanding. You may include this plot in your report if you wish to.
  - Use **`polyfit`** to estimate the *slope* and *intercept* of said plot.
  - You should get **`2*n_electrodes`** (i.e. $2 \cdot 19 = 38$) new features.
- Compute the *spectral moment*:
  - Treat the normalized power as a probability function and compute the mean frequency:

$$spectral\ moment = \sum_f p_{norm}(f) f$$

        - You should get **`n_electrodes`** (i.e. 19) new features.
    - Compute the *spectral edge*:
        - Compute the frequency at the edge of the top decile, i.e. the frequency that 90% of the power resides below it and 10% of the power resides above it:

$$spectral\ edge = f_e \text{ s.t. } \sum_{f<f_e} p_{norm}(f) = 0.9$$

            - Useful functions for this part: **`cumsum`**, **`diff`**, **`find`**.
        - You should get **`n_electrodes`** (i.e. 19) new features.
    - Compute the *spectral entropy*:
        - Treat the normalized power as a probability function and compute its entropy:

$$spectral\ entropy = -\sum_{f} p_{norm}(f)\log_2\left(p_{norm}(f)\right)$$

        - You should get **`n_electrodes`** (i.e. 19) new features.
- Note that each feature has its own units. To address that, standardize each feature by subtracting its mean and divide by its standard deviation. Use the function **`zscore`** (make sure to use it properly).
- Overall, you should extract **`n_electrodes*(2*n_freq_bands + 6)`** (i.e. 342) features from each time window.

4. PCA

- Perform PCA to reduce the data dimensionality from 342 to 3, so that you can visualize the data.
- Do this separately for each seizure.
- **DO NOT USE MATLAB's `pca` COMMAND!**
- Substract the mean sample from each sample.
- Find the covariance matrix of the data using the formula taught in class:

```
C = X*X'./(P - 1);
```

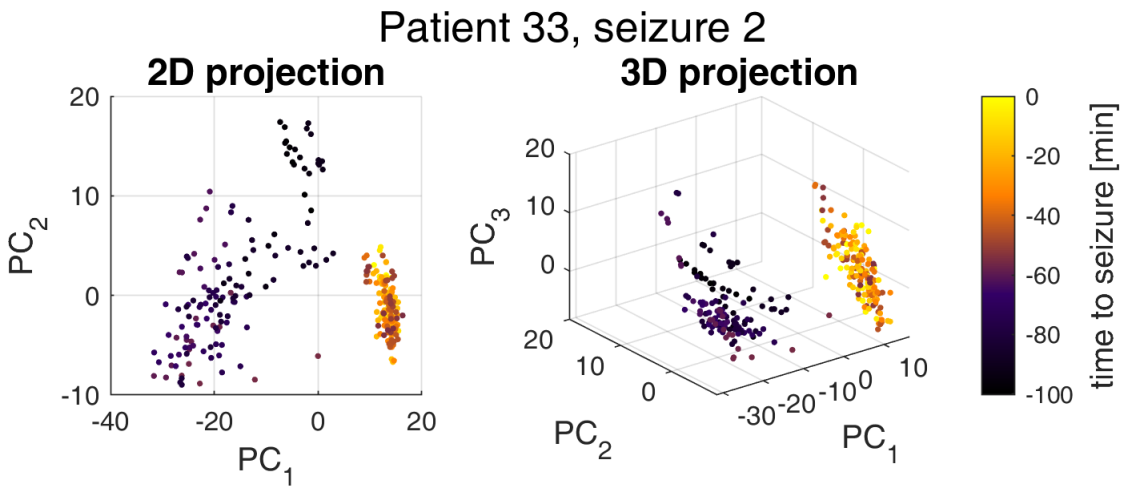
- Use either **`eig`** or **`eigs`** to diagonalize the covariance matrix. Explain your choice. (What are the differences between the two functions?)
- Use the formula taught in class to project the data onto the relevant eigenvectors.
- Useful functions for this part: **`diag`**, **`sort`**.

5. Plots

- Plot the data in 2D and 3D, using the first 2 and 3 first principal components (respectively). Useful functions are **`plot`**, **`scatter`**, **`plot3`**, **`scatter3`**.
- Add colors indicating the time to seizure (in **minutes**). Use the **`colorbar`** command.
  - Add a label to the color bar using the following syntax:

    ```
    cb.Label.String = 'colorbar label';
    ```

    where **`cb`** is the handle of the **`colorbar`** object.
  - Change the color scheme using the **`colormap`** function (you may choose whatever color scheme you wish).
- Write the patient and seizure numbers in the title of the figure (use **`sgtitle`** etc.).



Exercise deliverables

- Submit according to the submission guidelines in Moodle.
- You will hand in a .zip file (NOT .rar or any other compression file type), containing:
  - MATLAB code of the main script.
  - Any additional functions you wrote (**at least two**).
  - A report containing at least:
    - Brief description of the data.
    - Brief description of your work, including the parameters and features used. Do not forget to address any questions and special requests described above.
    - Requested figures – you should have 2 figures (2D and 3D) for each seizure.
    - **Your own** insights and conclusions.

Good Luck!!