Sleep Stage Classification for Infants and Children

Institute of Medical Informatics

University of Lübeck

Contents

- **Project Presentation:**
  - About Sleep Lab Project
  - Pattern Recognition Chain (PRC)
    - Sleep Data Acquisition
    - Dataset Description
    - Sleep Data Preprocessing
    - Classification and Comparative Analysis
- **Organization:**
  - Project Objectives
  - Organization of Scenario 3
  - Advised Checklist

Project Presentation

About Sleep Lab Project

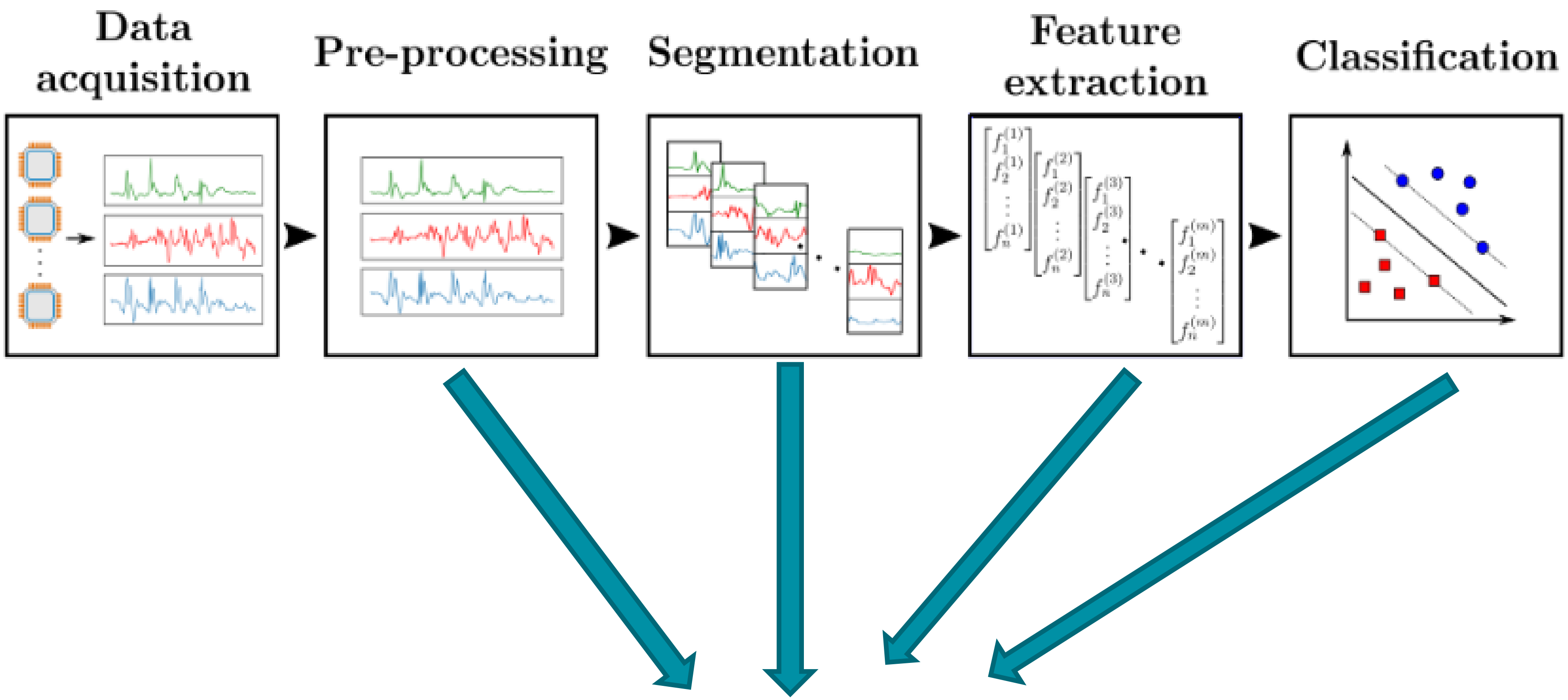
- In the sleep medical research area, effectively sleep stage classification can help the expert better understand and diagnose the sleep disorders of infants and child patients.

- **Data type:** Clinical sleep PSG recordings, such as EEG, EOG, EMG, etc.

- **Goal**: Investigate the relevance of the different sensor channels to the sleep stage classification for child patients.

- **Problem:** how to find relevant sensor modality to efficient classify the sleep stage?

Pattern Recognition Chain - PRC(1/11)



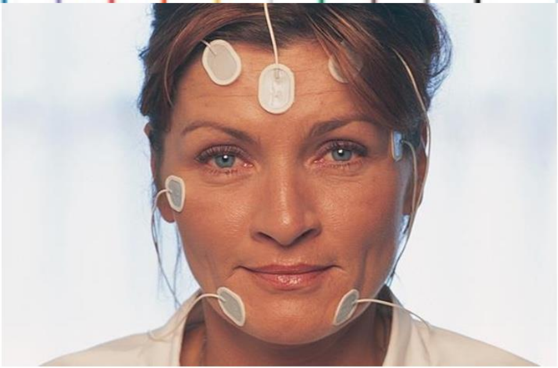
To be done in this project

PRC - Sleep Data Acquisition(2/11)

- Sensor device in sleep lab:
  - Hardware: Headbox with Alice 6 (multi-function sensor device system)
    → General sensor device used in many hospitals

Headbox

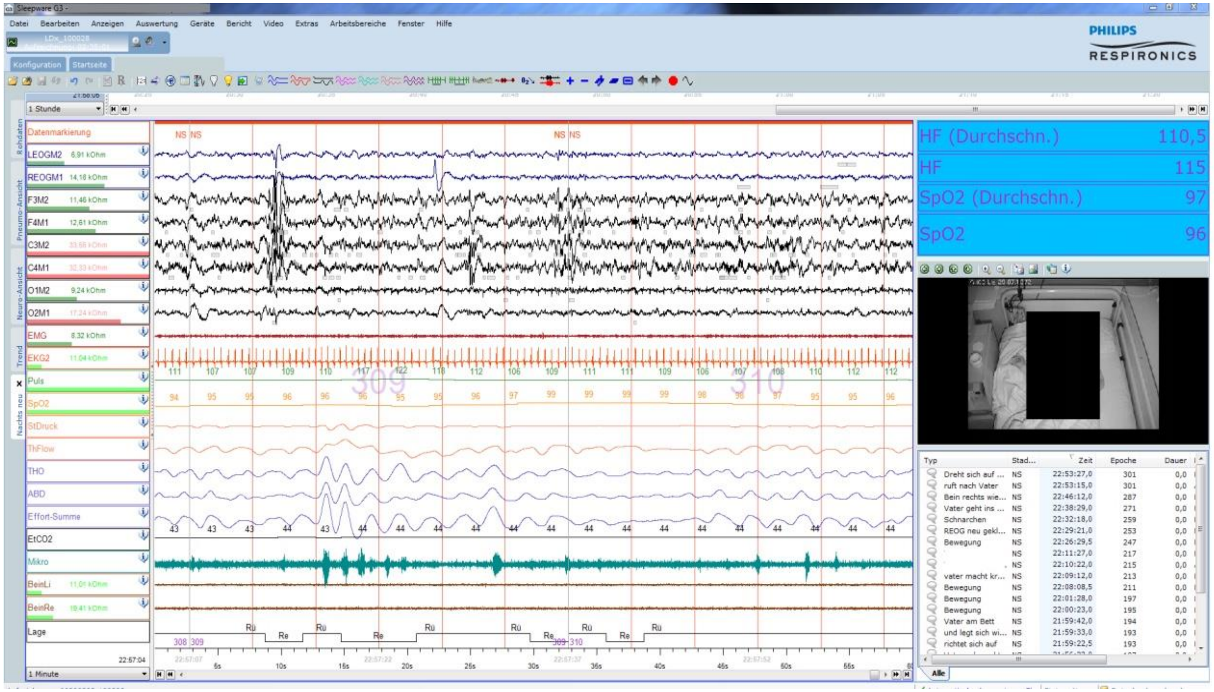
Ambu® Neuroline Cup electrode



Ambu® Neuroline-720 sensor sticker

PRC - Sleep Data Acquisition(3/11)

- Software:
  - This software system developed for Adults
  - Classifier is invalid for children's sleep data → **<u>Accuracy < 20%</u>**
  - Sleep expert score the labels manually → **<u>Time-consuming</u>**



Philips Sleepware G3

PRC - Dataset Description(4/11)

- **5 experimental subjects** aged from 5 to 7 years old;
- Each subject collects **4 different sensor modalities** a total of **11 sensor channels** and then be save in **.csv** files:
  - **6 EEG channels:** O1M2, O2M1, C4M1, C3M2, F4M1, and F3M2 depend on different detection areas over the head;
  - **2 EOG channels:** LEOGM2 and REOGM1 represent left eye and right eye movement information;
  - **1 EMG(chin muscle) channel:** EMG;
  - **2 EMG(legs muscle) channels:** BeinLi and BeinRe describe double legs' muscle information.

PRC - Dataset Description(5/11)

- Length of sleep monitoring data: around 10 hours from evening to the next day morning
- Sampling frequency: ***EEG → 200HZ, EOG → 150HZ, EMG(chin) → 100HZ, and EMG(legs) → 50HZ.***
- Labeling: Sleep stages of sensor channels labeled by the expert in the sleep lab according to the American Academy Sleep medical (AASM) rules → **Every 30 seconds score a label and saved in SleepStaging.csv**

Label:	WK	REM	N1	N2	N3
Sleep stage:	Wakefulness	Rapid Eye Movement	**Transition stage**	Light sleep	Deep sleep
Class proportion (%)	19.83	19.63	**5.55**	30.45	24.54

- Dataset provided on Moodle (2 types)

→ Sleep_data_SS2020: size **~ 3**33MB;

→ Sleep_data_downsampling_AllSensorChannel_lowfrequency_10HZ: size ~ 23MB.

PRC - Sleep Data Preprocessing(6/11)

- Experimental data selection:

Objective: To analyze, which sensor modality and sensor channels for the sleep stage classification are efficient?

* **Constraint**: Need to utilize all sensor channels and sensor modalities to train the baseline model.

* **Considering**:

1. The difference in sampling frequency
2. Size of input data
3. Selection of the classifier

PRC - Sleep Data Preprocessing(7/11)

- Preprocessing **suggestions**:
  - Data Augmentation (Noise Injection or Shifting, etc.)
  - Using timestamp as index to concatenate sensor data to create a hybrid modality
  - Sampling frequency normalization
  - **Maybe:** Downsampling (e.g., 200HZ→50HZ or even smaller) regarding to the computational power and the complexity of trained model

  *__Note:__ PSG recordings have different feature peak frequency bands. Excessive downsampling may result in the loss of valid information

→ **EEG frequency band**:

Alpha: 8 to 13 HZ

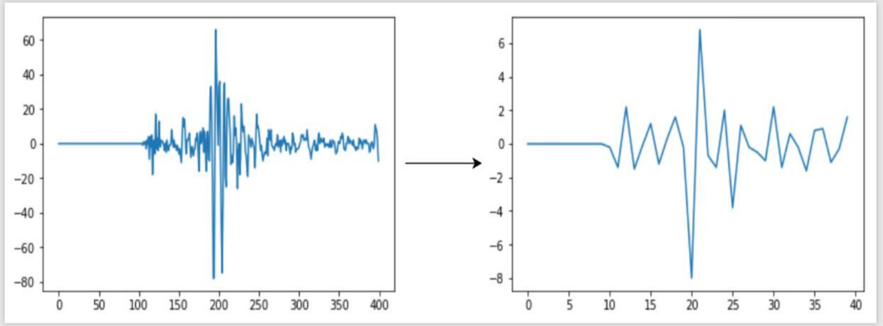
beta: > 13 HZ

theta: 4 to 7.9 HZ

delta: 0.5 to 3.9 HZ

→ **EOG frequency band**: 0.2 to 15 HZ

→ **EMG frequency band**: 15 to 45 HZ



Downsampling using 20HZ

PRC - Sleep Data Preprocessing(8/11)

- **Data Augmentation (DA):** Data Augmentation is a technique that can be used to artificially expand the size of a training set by creating modified data from the existing one. It is a good practice to use DA if you want to prevent overfitting, or the initial dataset is too small to train on, or even if you want to squeeze better performance from your model.

- **Class Partition Table:**

Label:	**WK**	**REM**	**N1**	**N2**	**N3**
Class proportion (%)	19.83	19.63	**5.55**	30.45	24.54

* The features of the N1 and REM stage in EEG are quite similar → misclassification !

- **Types of DA**
  - Images (Geometric transformations, Random Erasing, etc.)
  - Text (Word replacement, Word/sentence shuffling, etc.)
  - Time series (**Noise Injection**, Shifting, etc.)

PRC - Sleep Data Preprocessing(9/11)

- **Data Augmentation (DA):** Time series (**Gaussian Noise Injection**)

- **Advantages:** We get more data for our deep neural network to train on and can train our neural network on noisy data which means that it will generalize well on noisy data as well.

- **Disadvantages:** Noise can reduce the accuracy of neural networks.

- ***Gaussian Noise Injection:***

- Gaussian noise is statistical noise with the function of equal probability density in the normal distribution, also known as Gaussian distribution.

- The noise produced here is random, but like most random events, a certain pattern follows. The reason why Gaussian distribution is highly preferred in data science or machine learning is that Gauss random events are very common in nature.

- The Gaussian distribution probability density function is calculated by the following formula:

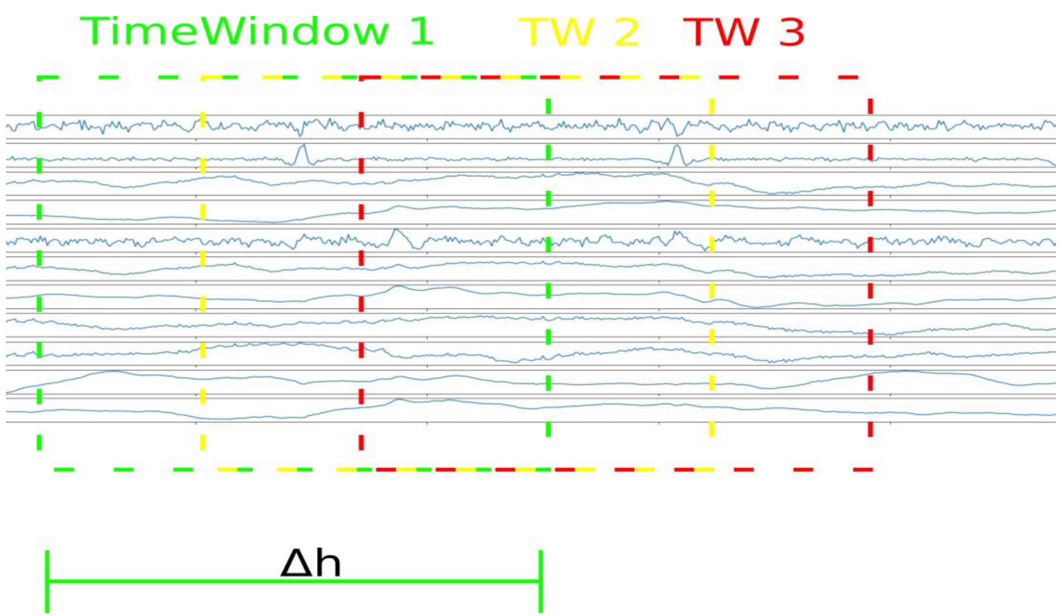
$$N(\chi : \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{1}{2}(x-\mu)^2 / \sigma^2)$$

- **Gaussian noise to a data set can be added as follows:**

1. Random noise is calculated and assign noise to the variable.
2. **Mean (μ) is set to zero and the standard deviation δ can be generated as needed. The standard deviation of the random noise controls the degree of dispersion (δ = 0.05, 0.2, ...).**
3. **Add the noise to the data set (Data set = Data set + Noise).**

PRC - Sleep Data Segmentation(10/11)

- Segmentation **suggestions**: Sliding window method
  - The size of time window can be set based on your own wishes, but according to the labeling rules, **minimal window size: TW = 30sec** for this sleep experiment
  - The simplest form: **TW = 30sec without window overlapping**
  - Window overlapping form **(not suggest)**: **TW > 30sec and sliding size (Δh) < TW** → consider the determination criteria of label



PRC – Classification and Comparative Analysis(11/11)

- Classification **suggestions:**

  - No constraints, free to choose the classifier
  - Use all sensor channels to train the classifier to set the baseline of the comparison
  - Select the targeted sensor channel (untargeted sensor channels set to zero) to train the baseline model

- Evaluation and comparative analysis **suggestions:** **(no constraints, free to choose)**

  - *F1-score*
  - *K-Fold* Cross-validation
  - Jacobian-Matrix
  - Cohens Kappa coefficient
  - ......

Project Organization

Project Objectives

- **Objectives**:
  - Sleep stage classification for child patient based on PRC
  - Solve the class imbalance problem
  - Analyze the relevance of the different sensor channels to sleep stage classification
  - * Improve the performance of the sleep transition stage (N1) ?
- **Constraints:**
  - You must implement data preprocessing (data augmentation)
  - Using all sensor channels and modalities to train the baseline model
- **Additional specifications:**
  - All classification approaches are allowed
  - All Python libraries are allowed

Project Organisation

- Project carried out in **groups of 2 or 3 (same as last scenarios or change)**

- Milestones:
  - **1st week**: overview of the whole solution and understand the sensor modalities and sensor channels and implement data preprocessing
  - **2nd week**: data augmentation and segmentation. Feature extraction if needed.
  - **3rd week:** train baseline model and evaluate the performance
  - **4rd week:** evaluate the relevance of the sensor channels to classification

Project Evaluation

For the third scenario about sleep stage classification:

- **Solution** = Python code which implements the preprocessing, data augmentation, and classification model training. Also, the code of evaluation of the relevance of sensor channels to classification.
- What is expected to be in the report at minima:
  - Description of the python implementation
  - Description of sleep data preprocessing, data augmentation method and classifier training
  - Results of the relevance evaluation of sensor channels to classification
  - Short explanation of the structure of the Python code
- The structure and length of the report are up to you.
- **Main evaluation criteria**: scientific soundness of the proposed approach.

Advised checklist for week 1

- Discuss group organization
- Download the raw dataset, understand the structure of data and discuss the solution
- Determine which preprocessing methods will be executed

*__Note__: keep in mind a training session can take a lot of time depending on the size of the training set, your computational power and the complexity of your model. If the training process takes too much time, you need to downsample the original data.

- Implementation of data preprocessing in Python

Advised checklist for week 2

- Check the literature about data augmentation
  - If you can find other useful method, understand how to use it
  - If not, implement Gaussian Noise Injection to extend the partition of the N1 stage
  - Just implement the simplest form of sliding window segmentation and ensure all labels are matched

- Feature extraction (if needed)

  * Well-known feature extraction in the sleep medical science:

    - Discrete Wavelet Transform (DWT)

Advised checklist for week 3

- Train your classification baseline model utilizing all sensor channels
- Evaluate the performance of your baseline model
- Start to determine how execute the evaluation of the relevance of the different sensor channels to the classification.

Advised checklist for week 4

- Utilize targeted sensor channels to train your classification model
- Compare the results with the baseline model to evaluate the relevance of different sensor channels to classification
- Write guidelines for your Python code
- Start with the writing of the report if not already done