Optimizing Sleep Stage Classification using MOIRA

Course: STAT 4830 – Numerical Optimization

Date: February 7, 2025

Project Code Name: STAT-4830-MOIRA

1. Project Objectives

- Develop an optimized **epoch-wise classifier** for sleep staging from **scalp EEG**.
- Improve **wake vs. N1 sleep classification**, particularly during **sleep onset transitions**.
- Address **limitations of existing classifiers**, such as:
  - Over-reliance on **in-bed intervals**.
  - **High sensitivity** to sleep epochs but **low specificity** for wake epochs.
  - **Generalization issues** across different subjects and time-of-day variations.

2. Dataset Description

- **Data Source:** EEG recordings from **29 subjects**, stored in **EDF format**.
- **Sleep Stage Labels:** Provided in **text files**, with **one label per 30-second epoch**.
- **Preprocessing:**
  - **Bandpass filtering (0.5–40 Hz)**
  - **Epoch segmentation (30s windows)**
  - **Normalization per subject**
- **Feature Extraction:**
  - **Power Spectral Density (PSD)** features for **delta, theta, alpha, beta** bands.
  - **Catch22 time-series features** capturing autocorrelation, entropy, and complexity.

3. Methodology

3.1 Mathematical Formulation of the Optimization Problem

Problem Setup:

Given an EEG time series (Y) with associated covariates (Z), we seek to predict the sleep stage (S) for future time steps using a learned function:

[
\hat{S}*t = f*{\theta}(Y_{t-l:t}, Z_{t-l:t+h})
]

Objective Function:

- We train (f_{\theta}) to maximize the **log-likelihood** of the predicted sleep stage distribution:

[

3.2 Implementation using MOIRA

MOIRA: Masked Encoder-based Universal Time Series Forecasting Transformer

- Why MOIRA?
  - **Handles multivariate time series** effectively using a **masked encoder architecture**.
  - **Learns from unobserved data (zero-shot learning)** to improve generalizability.
  - **Uses multi-patch projections** for capturing hierarchical temporal structures.
  - **Outputs probabilistic distributions** rather than single-point predictions.

MOIRA-Based Classification Pipeline

1. **Preprocess EEG Data**
   - Load EDF recordings and segment into **30-second epochs**.
   - Apply **bandpass filtering (0.5–40 Hz)**.
   - Extract **PSD features + Catch22 features** for each epoch.
2. **Feature Transformation for MOIRA**
   - Flatten multivariate EEG time series into a **masked encoder format**.
   - Apply **multi-patch projection layers** to generate vector embeddings.
   - Use **learnable masked embeddings** to encode future sleep stage predictions.
3. **Training Objective**
   - Train the model using **cross-entropy loss**:

4. Preliminary Results

- **Implemented MOIRA on a subset of EEG data (first 10 minutes per subject).**
- **Observations:**
  - Initial classification accuracy: **78% (wake vs. N1)**
  - **Improvements over traditional PSD-based models**: Better separation of **relaxed wakefulness** from **N1 sleep**.
- **Bottlenecks Identified:**
  - **Sequential feature extraction** is slow → need parallelization.
  - **Model fine-tuning for generalization across subjects**.

5. Progress and Next Steps

Current Bottlenecks

- **Long Processing Time:**
  - Catch22 feature extraction is computationally expensive.
  - Running full 24-hour EEG recordings is currently infeasible.
- **Generalization Issues:**
  - Model needs domain-adaptive fine-tuning.
  - Requires evaluation on external datasets.

Planned Improvements

- **Parallelization Strategies:**
  - Implement `joblib` or `multiprocessing` to speed up **feature extraction**.
- **Deep Learning Extensions:**

6. Repository Structure

Code Organization

Documentation Strategy

- **README.md** → Project overview, installation, and dataset description.
- **Inline comments** → Clarifying key functions in feature extraction and MOIRA training.
- **Jupyter Notebooks** → Exploratory data analysis and baseline model comparisons.

Conclusion

- **MOIRA has demonstrated initial improvements** in wake vs. N1 classification.
- **Further tuning and parallelization** are needed to scale to 24-hour EEG recordings.
- **Next steps involve deep learning integration** to enhance robustness and generalizability.