Introduction to Physiological Signal Processing and Learning

Signal Analysis and Interpretation Lab, USC

March 19, 2024

1 Introduction

SAIL is extensively conducting human subjects research using computational methods. Working on interdisciplinary threads and projects typically requires a combination of technical knowledge and scientific grounding, as well as abiding by strict ethical and privacy-related guidelines. This is why you have completed the associated CITI training and assigned a SAIL server to work on our data. In this assignment you will be introduced to a typical pipeline for analyzing such data, and you will be guided to apply common research practices, such as coming up with assumptions and validating them experimentally using signal processing and machine learning tools.

2 Assignments

Below you will be introduced to a series of tasks that you should visit in order. These tasks are designed to help you familiarize yourself with basic statistical concepts and machine learning pipelines, and also provide a kick start in analyzing physiological signals, with a focus on cardiac activity.

2.1 Machine Learning Fundamentals

For our purposes, machine learning (ML) is the process of developing models to learn patterns or make predictions based on data. Although this is not a ML assignment, basic familiarity with the concepts outlined below is expected. You can always refer to this textbook or any ML textbook for more details. A Machine Learning task typically involves the following components:

1. **Dataset -** a set of samples (e.g., images, audio snippets, text documents, etc) from which an ML model can learn. In our case, the dataset that we will work on consists of physiological signals of human subjects, along with labels describing their mental and experimental state. This is an example of *supervised learning*, where each sample is accompanied by a single or multiple labels that indicate desired attributes we want to model. For example, an image dataset curated for supervised learning would contain image samples along with labels describing the content of the image (e.g., 'dog' or 'cat'). Recently, and due to the challenge of obtaining labels for large amounts of data, advances have been made in *unsupervised* learning, where ML models are trained to learn representations solely from the data structure itself.

2. **Objective -** the task which an ML model is trained to perform. In our case, we would like to automatically estimate mental states such as stress and emotion, given physiological signals as input. In supervised setups like this, the objective can be viewed as a *classification* task, where the model is trained to select the best discrete label that fits the input data (e.g., 'dog' or 'cat'), or as a *regression* task, where the model is trained to estimate a continuous value (e.g., the expected rent price of an apartment). Typically, ML models are evaluated using a set of loss functions and accuracy metrics that are suitable to the task in hand, e.g., accuracy, recall, or F1 score for classification, and mean squared error, or correlation coefficients for regression. Evaluation should always be done on a distinct partition of the dataset (*test set*) for which no information is embedded in the model, hence there is no 'cheating' in measuring model performance.

3. **Feature Extractor -** a series of signal processing steps to preprocess, clean the input samples and compute feature values that can effectively discriminate between different output labels.

This step is essential since the data provided in most cases cannot be used directly to estimate the target. For example, the physiological signals in our dataset contain invalid samples, missing data, and noisy recordings that should be excluded. Further, the length of the recordings is uneven and comes from different subjects, hence the data should be sliced and normalized before used to train an ML model. Feature extraction is traditionally a manual process, where engineers select and compute features based on domain knowledge. For example, to model cardiac activity for stress detection, most works extract *Heart Rate Variability* (HRV) features, which have been shown to be sensitive to variable stressors. Recently, this whole process has been replaced by automatic feature extractors, built using neural networks (*representation learning*).

4. **Model -** the ML algorithm used to estimate the target of interest, e.g., classify input data to their corresponding labels. A wide variety of models have been suggested over the years, both for *shallow learning* (i.e., operating on already computed feature vectors), and *deep learning* (i.e., operating on minimally processed signals and incorporating the feature extraction stage described above). Typical shallow classifiers are *Naive Bayes, Support Vector Machines,* and *Random Forests*. Deep learning models are neural networks (NN) and have dominated the field of data analytics due to their superiority and robustness in performance. Typical NN architectures used today are *multi-layer perceptrons* (MLP), *convolutional neural networks* (CNN), *recurrent neural networks* (RNN), and *Transformers*. Training deep learning models usually requires a substantially large labeled dataset. For now we will focus on simple shallow classifiers.

Each subsection below will delve into each of these 4 categories, using WESAD as an example dataset of physiological signals. In the last section we will discuss inference made from the ML experimentation. To drive through the accompanied Python scripts and Jupyter Notebooks, clone the repository from https://github.com/klean2050/SAIL-onboarding and follow the instructions from `README.md` to set up your environment in the server. You should not push any changes you make.

2.2 WESAD Dataset Exploration

The dataset for *WEarable Stress and Affect Detection* (WESAD) contains physiological data from 15 participants. RespiBAN Professional sensors were used to collect data at a sampling rate of 700Hz. The goal was to study four different affective states (*neutral*, *stressed*, *amused*, and *meditated*). First, 20 minutes of neutral condition data were collected, during which participants were asked to do normal activities. Then participants watched 11 funny video clips (amusement) and went through public speaking and arithmetic tasks (stress). Finally, they went through a guided meditation session of 7 minutes. Upon completion of each trial, labels for the affective states were collected.

WESAD tracks the affectove state of the participants through multiple modalities, including ECG, BVP, EDA, EMG, respiration, skin temperature, and acceleration. In this assignment we will focus on the cardiac activity, as it is recorded from the ECG sensor. Go through the `ECG_Intro_WESAD.ipynb` notebook and complete the code to load, visualize, and pre-process the ECG signals.

2.3 Useful Measures of Cardiac Activity

The ECG signal itself is not very informative. Its time-series representation that you inspected looks repetitive and a simple model would struggle in locating characteristics of interest without proper feature extraction. In clinical settings, human experts typically evaluate ECG from 12 leads, which can provide nuanced information regarding cardiac rhythm, conduction abnormalities, presence of ischemia or infarction, and others. This information helps in diagnosing various cardiac conditions such as arrhythmias. In Psychology and Affective Computing, researchers are more interested in *heart rate*, i.e., the average number of heart beats per minute (BPM), and *heart rate variability* (HRV), i.e., the temporal variation of RR interval length. These measures are considered robust markers of stress, arousal, mood changes, and can indicate mental disorders like depression.

To assess heart rate activity, one needs to first locate the R peaks in ECG and then compute appropriate statistical measures. Through the `HR_Feature_Extraction.ipynb` notebook you will build the tools to detect R peaks in ECG signals, compute heart rate measures, and verify your implementation.

2.4 Assumption and Experimental Setup

A well-established assumption is that heart rate increases during stressful periods, due to the activation of the sympathetic nervous system. When faced with stress, the body releases stress hormones that stimulate the heart to beat faster, preparing the body for action in response to perceived threats. Conversely, during resting times, the parasympathetic nervous system becomes dominant and slows down heart rate. It is activated to promote restorative processes and conserves energy for the body. Moreover, HRV is heavily influenced by the balance between the sympathetic and parasympathetic systems. During periods of stress, HRV tends to decrease, as the sympathetic nervous system dominates and leads to a more consistent, less variable heartbeat pattern. On the other hand, HRV increases at rest, reflecting the body's ability to regulate changing conditions and neural effects.

We will test those assumptions on WESAD by setting up a Machine Learning experiment. As a reminder, WESAD involves distinct phases that correspond to different affective conditions (*neutral*, *stressed*, *amused*, and *meditated*). Here we will attempt to differentiate between *neutral* and *stressed* periods, based on the assumptions we already discussed. Go through the steps in `ECG_ML_WESAD.ipynb` to gather the features of interest and set up a simple ML classifier.

2.5 Machine Learning Validation

It is time to train our ML framework on the data we got from WESAD. In the same notebook, go through the second batch of steps to run the cross-validation setting and get the session predictions for each participant. You will then have to evaluate the predictions based on set of classification metrics that you will build. Keep notes of the results and insights you get from each of the metrics.

3 Conclusion

This assignment was designed to be an informative introduction to physiological signal processing and learning, with particular focus on evaluating cardiac activity for behavior analysis. At the beginning you were exposed to a summary of fundamental properties and pipelines for building Machine Learning frameworks. By finishing the steps outlined above, you will have gained a holistic understanding of the experimental process: getting to know and cleaning your data, formulating the feature extraction process, laying down assumptions, and testing them using statistical and learning algorithms.

However, research is not an assignment, and most of the times we come up with bad results, unexpected outcomes, and insights that require explanation and verification. In the task you implemented here, you probably identified some of those issues. Moving forward, it is important to document this feedback, provide explanations and, if necessary, go back and adjust your experimental process.