Annotated Bibliography of Current Research

Shaun Rask

October 2024

1 EEG Normal Waveforms

https://www.ncbi.nlm.nih.gov/books/NBK539805/
This article explains how EEG activity reflects the temporal summation of synchronous activity between millions of cortical neurons. It emphasizes the importance of gathering patient information before conducting analysis, as confounding variables such as age, level of consciousness, and environmental stimuli can all affect the waveforms. The article describes the common frequency ranges in EEG, using Greek letters to categorize them: delta, theta, alpha, sigma, and beta, in increasing order. Additionally, it highlights common tasks that each frequency range represents in the brain. It also discusses EEG transients, which are distinguishable waveforms or complexes primarily seen during drowsiness and light sleep.

2 Measuring Brain Waves in the Classroom

https://www.repereong.ro/articole-cursuri/measuring-brain-waves-classroom
This article goes over what brain waves are, how they are measured in the lab and in the classroom, as well as the significance of those measurements. Put simply, brain waves are neurons that communicate with each other when the go up and down, resembling waves. We measure these waves using a technique known as electroencephalography, where small electrodes are placed on a person's head and detects the brain waves. These waves vary in speed, with the frequency measuring different mental states such as tired or awake. The way to examine these brain waves further are though ERPs (event-related potentials), which are the electrical brain response to specific events, such as

reading a word or controlling an impulse. To use ERPs, a participant must perform a task that is designed to study a particular function of the brain.

3 How Deep Learning is Changing Machine Learning AI in EEG Data Processing

`https://www.bitbrain.com/blog/ai-eeg-data-processing`
This article discusses the strengths and uses of EEG systems, highlighting the challenge of interpreting EEG data, as results can vary significantly between individuals. Machine learning and AI offer the potential to automate and improve EEG data analysis, making it more applicable in Brain-Computer Interface (BCI) applications. The article outlines the three major steps in EEG data processing: pre-processing to handle noise and artifacts, feature extraction to generate descriptors for decoding tasks, and the final step of decoding, where models classify and transform EEG features into high-level signals such as letters, motions, or cognitive states. Deep learning models, particularly convolutional neural networks (CNNs), autoencoders, and recurrent neural networks, have been effective in capturing complex EEG data patterns. This image shows that CNNs are the most commonly used architecture, with autoencoders and recurrent networks used as well. However, challenges remain, such as the high cost, time requirements, and limited availability of EEG datasets. Training deep learning models efficiently for EEG data is still an ongoing area of research.

4 Personalizing an AR-based Communication System for Nonspeaking Autistic Users

`https://dl.acm.org/doi/pdf/10.1145/3640543.3645153`
This research paper explores the development of a personalized Augmented Reality (AR) communication system for nonspeaking autistic users. The system is based on a virtual letterboard, with automated positioning to assist users in spelling out words. Data is collected from the user's interactions with a physical letterboard, which is then used to train a machine learning model. Once the model is trained, it is integrated into the AR system, where a virtual letterboard is customized for the user. The paper provides an in-

depth discussion of the neural network setup and experiments conducted to validate the system.

5 An innovative P300 speller brain–computer interface design: Easy screen

`https://www.sciencedirect.com/science/article/pii/S174680942200115X`
This article highlights the P300 speller, which are BCIs that display desired characters/words once at a time using ERPs (event related potentials) with the response of flashing visual stimuli to determine the outcome. The interface used mostly involves a 6x6 or 7x7 letter matrix, named the easy screen P300 speller, where the rows and columns of the matrix flash and fade in random order, and when the flashing row/column connects to the target character, the subject is warned by the flash and the ERP is detected 300ms after the character is illuminated. Many different designs have been tested, using different dimensions, shapes, and flash times. There are multiple machine learning algorithms that can classify emotion such as a support vector machine (SVM), k-nearest neighbors (k-NN), and many more.

6 Machine Learning Methodologies in Brain-Computer Interface Systems

`https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=459eb1d35e2b394b71c`
`~:text=P300%20Speller%20is%20a%20BCI,signals%20after%20about%20300%`
`20ms.`
This paper attempts to evaluate the performance of different machine learning algorithms to find which has the highest classification accuracy. They go over the process of implementing each machine learning algorithm on the BCI dataset as well as the results they found from each. They found that SVM and BLDA algorithms yielded the highest accuracy for their 3 subjects. There were many issues such as low signal-to-noise ratio of EEG signals and a variance between each different subject. During the feature extraction phase, they found the appropriate feature extraction phase for each subject which helped fix the variance. For the classification phase, it is important to use linear classifier algorithms over linear classifiers, for example using linear

SVM classifiers instead of linear SVM.

7 Info on P300 speller

https://doi.org/10.1007/s10479-020-03921-0 A P300 speller consists of 36 alpha-numeric characters in a 6x6 matrix, the user has to focus their attention on a character, then rows and columns will flash in a random sequence. There are many different variations to the P300 paradigm that try to improve the framework. In the classical variation, a sequence of 12 different flashes (6 rows and 6 columns) is called an iteration. These flashes are an oddball paradigm, so there are two different kinds of stimuli, target and non target, occurring at frequencies of 0.166 and 0.833 respectively. The target response which is more rare should elicit the P300 response, which in this case the target stimulus is the row/column containing the desired character. We need a classification algorithm to distinguish between these stimuli. In ERP-based BCIs, to perform a selection step, many iterations are needed to improve the SNR (signal to noise ratio). Machine learning and optimization can be used to improve the classification performance.

8 Support vector machines

https://doi.org/10.1016/j.dcn.2022.101096 Classifier learning can be seen as finding a curve that well separates two sets of points. We are trying to find an optimal curve out of infinitely many curves. It's difficult to define optimal, because if we say that optimal is the curve that minimizes mistakes in the two sets, then it might not work as well on unseen data, which is known as overfitting. SVMs help avoid overfitting by finding the points in the dataset closest to the line separating the two classes; these are support vectors. Then the SVM will maximize the margin (distance) between the vectors and separation line. After it finds a line using the margins, it will create a formula, called the weight vector, that shows how important each feature is in deciding the classification. Then when you need to classify an object, it will plug its features into the weight vector and get a score, if the score is positive then its one class and negative will be the other. Before this training, it's important that the different features are on a similar scale so the model is not biased by a feature with large numbers. This is done by subtracting the mean and dividing by the standard deviation.

9 Importing MATLAB Files into Python: A Step-by-Step Guide for EEG Data Analysis with MNE

https://medium.com/@neurotist/importing-matlab-files-into-python-a-step-by-step-g
When loading EEG data from .mat file into python, the contents of the file will go into a dictionary-like object to hold each key-pair value in the EEG data. EEG data is usually organized as a multi-dimensional array with each representing channels, time samples, and trials using the shape attribute to show the size of each dimension. Then we need a MNE data structure to enable analysis and visualization. Using the MNE library, we create a information object which will hold the channel names, sampling frequency, and channel types. Note: The sfreq variable should be changed depending on your sampling rate, in the BCI competiton III dataset, the sampling rate is 512 hz.

10 Filtering EEG Data

https://neuraldatascience.io/7-eeg/erp_filtering.html EEG signals have many different frequencies, most coming in the range of 1-30Hz, where revelant activity happens in terms of EEG research. However there is a lot of unwanted noise picked up by an EEG which come in two forms, low and high frequency noise. Low frequency noise such as slow head movements or wire shifts occur in slow drifts on EEG data. High frequency noise like muscle movement or electromagnetic interference appears as a sharp change in the signal. Our job when filtering is to get rid of noise outside of the desired range. During recording its important to avoid aliasing which happens when high frequency noise isn't captured correctly due to a sampling rate being too low. This leads to misleading low frequency artifacts so we must avoid this by finding the highest frequency that is reliable, for a 500 hz sampling rate, about up to 167 hz is reliable. The best way to implement this is to use a low-pass filter to remove the high frequency range and a high-pass filter to eliminate the slow drifts.

11 Visualizing EEG data in python

`https://neuro.inf.unibe.ch/AlgorithmsNeuroscience/Tutorial_files/DataVisualization.html` When we begin visualizing an EEG dataset, the first step is to look at a single trail. After inspection, we can look through all the recorded epochs to evaluate the quality of the dataset. The easiest way to visualize an EEG dataset is to use the plot() function using MNE, which plots epochs. The y axis shows the channel names while the x axis shows the epoch numbers. We can also see the ratio of each event type above the plot. We can also plot average epochs to focus on the average epoch response, ERPs. This technique is used widely in the EEG field of research as it gets rid of irrelevant responses to a given task. The average ERPs only contain activity that appears at consistent latencies and electrode locations across repititions. There are many other things we can plot such as topographic information and channels but we will stick to the basic plots for now.

12 Time and Frequency Domains

`https://neuraldatascience.io/7-eeg/erp_filtering.html` For EEG data used in a P300 speller system, it is best viewed in the time domain, as P300 is an ERP that appears as a distinct positive deflection around 300 ms after a stimulus. For P300 detection, EEG is typically filtered in the 1–30 Hz range to remove low-frequency drifts and high-frequency noise. This filter range aligns with focusing on the delta, theta, alpha, and beta bands, which represent typical brain activity related to cognitive processing without excess noise. For optimal results, a bandpass filter around 0.1–30 Hz is recommended for P300 data. In addition, the EEG data should have a high enough sample rate to reliably capture the P300 and associated frequencies. According to the Nyquist theorem, the sample rate should be at least twice the frequency of interest. For example, since we are interested in 30 Hz, a minimum sample rate of 60 Hz is required, though 250–500 Hz is frequently used for greater accuracy. Fourier Transform (FFT) and Power Spectrum: Understanding power spectral density (PSD) can simplify assessing noise levels at various frequencies. For instance, a 60 Hz power line artifact might show up as a spike in the power spectrum, indicating a need for filtering or artifact removal. You can verify the effectiveness of filters and identify any residual noise that could interfere with P300 detection by using the Fast

Fourier Transform (FFT) to examine the signal's frequency content.