Brain Disorder Analysis and Classification Using Tensor Representation of EEG Signals

By applying higher order extensions of linear discriminant analysis and regression

J.A.J.M. Vrijdag

TU Delft

Brain Disorder Analysis and Classification Using Tensor Representation of EEG Signals

By applying higher order extensions of linear discriminant analysis and regression

by

J.A.J.M. Vrijdag

to obtain the degree of Master of Science

at the Delft University of Technology,

to be defended publicly on August 23, 2024 at 10:00.

Student number:	4719026	
Project duration:	December 4, 2023 – August 23, 2024	
Thesis committee:	Dr. B. Hunyadi,	TU Delft, chair
	Dr. ir. R. C. Hendriks,	TU Delft, core member
	Dr. ir. J. L. Vroegop,	Erasmus Medical Center, external member
	Dr. ir. K. Batselier,	TU Delft, core member

An electronic version of this thesis is available at http://repository.tudelft.nl/.

TU Delft

Abstract

At the child brain facility at the Erasmus Medical Centre, multiple tests are performed with children who have one of several disorders. Two of these tests are done with electroencephalogram measurements and are called mismatch negativity and acoustic change complex. After a signal processing pipeline, the EEG measurements from these tests are shown as waveforms called event-related potentials. The goal of these measurements is to see if there is any relation between the waveforms and the disorder, age and other information about the subjects.

These waveforms are measured for each subject, EEG electrode, and in the case of MMN, for different stimuli, which naturally results in a tensor data structure. Algorithms for discriminant analysis and regression that are developed to be applied to tensors are described and altered to take into account the properties of the EEG data. Discriminant analysis can be used to improve classification algorithms that distinguish disorders, while regression can be used to predict variables such as test scores based on the measured data. The algorithms are first tested on simulated data, which shows they can have some merit. Classification rates improve in most simulated cases when the discriminant analysis is applied to the data. Regression can also reliably predict variables when strong correlations are present between the input tensor and output variable. Based on the data from the child brain facility, the discriminant analysis still improves classification rates in some cases, but not as significantly as on the simulated data. Regression using the algorithms described in this thesis is not useful in predicting test scores from other experiments done with the subjects.

The algorithms are also dissected to discover which specific features in the data tensor are weighted heavier by the algorithms. This is done to gain new insights into the differences between the disorders. When comparing the weights that are used for the simulated data with the features that are of importance, there is some relation, but not a very strong one. When the input tensor is however segmented in the time mode, the times of interest can be identified. The regression algorithm also resulted in weights that can be analysed to look at when and where the measurements relate to the output, but this did not show any interesting results. For real data, the segmented tensor resulted in some interesting insights about the differences between the various disorders.

For the current dataset, the discriminant analysis algorithms do improve classification rates, but not by much. The features weighted the most by this algorithm in combination with a segmented tensor might give some insight into the disorders. The tensor regression methods do not work to predict a test score and do not give new insights into the disorders.

Acknowledgements

I would like to thank all my supervisors, Bori, Richard and Jantien, for the support given during this thesis. Making a thesis is a far more elaborate research project than I have ever encountered during my studies, and doing it all by myself felt quite intimidating. Thanks to the supportive and useful feedback given by my supervisors, I quickly felt more comfortable tackling this project. When I was stuck with a problem, Bori and Richard often gave suggestions that helped me work past those hurdles. In analysing the ERPs, which is not my area of expertise, I got a lot of help from Jantien and from Marloes Adank as well.

My friends, roommates and girlfriend deserve a “thank you” as well for proofreading (parts of) my thesis and making sure I still had more going on in my life than just research papers, Matlab and LaTeX.

Finally, I also have to acknowledge Grammarly, which not only spellchecked my entire thesis but also helped me rephrase or elaborate on text using generative AI when I was having trouble making myself clear.

J.A.J.M. Vrijdag
Rotterdam, August 2024

Contents

Abbreviations and Mathematical Notations

Abbreviation	Definition
ACC	Acoustic change complex
BO	Brain overgrowth syndrome
BSHODA	Block sparse higher-order discriminant analysis
BSS	Blind source separation
CPD	Canonical polyadic decomposition
DA	Discriminant analysis
EEG	Electroencephalogram
EMC	Erasmus medical center
ERP	Event-related potential
GA	Grand average
GEVD	Generalized eigenvalue decomposition
HODA	Higher-order discriminant analysis
ICA	Independent component analysis
i.i.d.	Independent and identically distributed
KHL	Kinderhersenlab
KNN	K-nearest neighbours
LDA	Linear discriminant analysis
ML	Machine learning
MMN	Mismatch negativity
MS	Multiple sclerosis
MSE	Mean square error
PCA	Principal component analysis
PS	Persistent stuttering
RBF	Radial basis function
SB	Spina bifida
SCPD	Sparse canonical polyadic decomposition regression
SHODA	Sparse higher-order discriminant analysis
SNR	Signal-to-noise ratio
STD	Sparse tucker decomposition regression
SVD	Singular value decomposition
SWS	Sturge-weber syndrome
TD	Tucker decomposition
WT	Wavelet transform

Mathematical notation	Definition
x	Scalar
$\mathbf{x}$	Vector
$\mathbf{X}$	Matrix
$\underline{\mathbf{X}}$	Tensor

Mathematical notation	Definition
vec($\underline{\mathbf{X}}$)	Vectorization of tensor
$\mathbf{X}_{(k)}$	mode-k-matricization
$\times_k$	mode-k-product
$\times_k^l$	mode-k,l-product
$\|\|\underline{\mathbf{X}}\|\|_F$	Frobenius norm
$\langle \underline{\mathbf{X}}_1, \underline{\mathbf{X}}_2 \rangle$	Inner product

1

Introduction

One of the least well-understood parts of the human body is the brain. Looking like a slimy, simple grey mass from the outside, for most of human history the fact that this object controlled the entire body wasn't even known. This changed during the Enlightenment when science got a more central role in society. Slowly, humanity began chiseling away at the enigma surrounding our most vital organ by discovering new techniques to study it. One of these new discoveries that allowed us to get completely new insides on the activity within our grey mass was the electroencephalogram (EEG) in 1929 [1]. This measurement from the brain was obtained by placing electrodes on the scalp. Although it was first dismissed as being random correlated noise due to the small and seemingly noisy signal it measured, over the years this became a vital tool for understanding certain brain processes.

At the Kinderhersenlab (KHL), or child brain facility, in the Erasmus Medical Center (EMC) in Rotterdam, they aim to better understand various disorders in children, and the underlying brain processes. This is done by subjecting children to all kinds of experiments and tests in a playful environment, resulting in a broad range of data. A few of the experiments that are carried out involve an EEG measurement. Two of which are based on eliciting an auditory event-related potential (ERP). This ERP is a waveform in the brain that is caused by a stimulus, which is a sound in the case of the KHL. The two experiments using this sound are the mismatch negativity (MMN) and the acoustic change complex (ACC). Previous literature has shown that both the MMN and the ACC waveforms can show differences in how auditory information is processed [2–6]. The goal of this project is to use the ERPs from these experiments to be able to classify the disorders based on the EEG and get a better understanding of how these disorders work and develop.

1.1. Problem Statement

The ERP waveform is a broadly researched phenomenon, implying there is already a lot of knowledge on how to study evaluate ERPs [7]. The classical way of investigating ERPs, however, is based on fairly simple features of the waveform, such as amplitude and latency of events. The complete complex underlying behaviour of the brain is not captured by only these two numbers. By combining both clever signal processing and simple machine learning (ML) tools, the goal is to uncover more information on disorders than was previously possible. This goal can be more comprehensibly stated in the form of the following research questions:

1. How can novel signal processing techniques and machine learning involving tensors be used to classify and evaluate different disorders and their development in children?
2. What can such an algorithm tell us about the underlying processes that differentiate the various types of disorders?
3. How can these methods be applied to the real-world data from the Kinderhersenlab, and how do they perform in both classifying different disorders and uncovering information on related brain processes?

By answering these questions satisfactorily, the KHL can hopefully gain a better understanding of the children's auditory and speech processing and the relation to the different disorders. If the machine learning model's outcomes show high accuracy, it may even be used to make better predictions on the further development of cognitive functions related to some children's disorders.

1.2. Outline

This thesis begins with some important background information in Chapter 2. This background information is about the measurements made at the KHL, the signal processing that is done on the measurements and some techniques that can be useful to analyse the data. In Chapter 3 the resulting plots from applying the signal processing pipeline to the KHL data are shown and analysed. More advanced methods of analysing this data are described in Chapter 4. These methods are tested on simulated data, which is described in Chapter 5, and on the real data, which is described in 6. All Chapters will be discussed in Chapter 7, along with some suggestions to improve on the work in this thesis. Finally the thesis is summarized in Chapter 8.

2

Background

2.1. The Event-Related Potential and Electroencephalogram

An ERP is a very useful pattern that can be present in an EEG. The EEG is electrical activity measured in the brain by placing electrodes over the scalp. The signal corresponds with an electric field generated by groups of pyramidal cells that release neurotransmitters with either a positive or negative charge. The activity of one such cell is not measurable, but when many of these cells are active at the same time and produce the same electric field, the resulting potential can be picked up by one or more electrodes. In most cases, the EEG is measured using a number of electrodes ranging between 16 and 256. These electrodes are placed all over the scalp to get a sense of where the EEG activity is happening. Pinpointing the exact brain region is, however, very hard due to interfering signals and uncertainty on the direction of the electric field. The power of measuring using an EEG does, therefore, not lie in localizing phenomena, but in its high temporal resolution. Most EEG recorders have a sample rate of $> 100Hz$, and the electric field travels almost instantaneously from the generation site to the electrode. This property is often used in ERP research.

An event-related potential is a recorded potential in the brain that is generated by the presentation of a stimulus. The exact behaviour of the brain depends on the type of stimulus and whether or not a response to the stimulus is necessary. This brain activity of interest does have a low amplitude, so from a single trial, it is therefore not visible among other brain activities. This problem is solved by recording many ERPs, often a few dozen, and averaging them. Brain activity not caused and time-locked by the stimulus should cancel out, revealing the ERP waveform of interest. In general this results in a waveform that contains several peaks and valleys called ERP components. The amplitude and latency after the stimulus of each of these components are what is most often analysed in an EEG study, and are therefore very important. Each component, therefore, has a name, which often consists of a letter and a number. The letter indicates whether the component is a positive going (P), negative going (N) or stimulus-dependent (C). The number can indicate one of either two types of values, higher numbers (> 10) often indicate after how many milliseconds the component peak most often occurs while lower numbers ($1 - 10$) indicate if it is the n^{th} valley or peak. Some of these components also have subcomponents, which are specified with different names. To isolate a certain component of interest, a difference waveform can be used. Such a waveform is created by subtracting the ERP of different experimental setups. One of the methods that results in such a difference waveform is the mismatch negativity (MMN) paradigm.

2.1.1. Mismatch Negativity

An experiment and resulting waveform that is often used in ERP research is the mismatch negativity. Its effects were first discovered by Näätänen in 1978 [8]. This experiment is done by presenting a series of sounds to a subject. Most of these sounds are the same, but every so often an oddball stimulus is presented. This can be either a sound that has a different frequency, amplitude or length. These oddballs create a more negative potential in the ERP [8]. The mismatch negativity is obtained

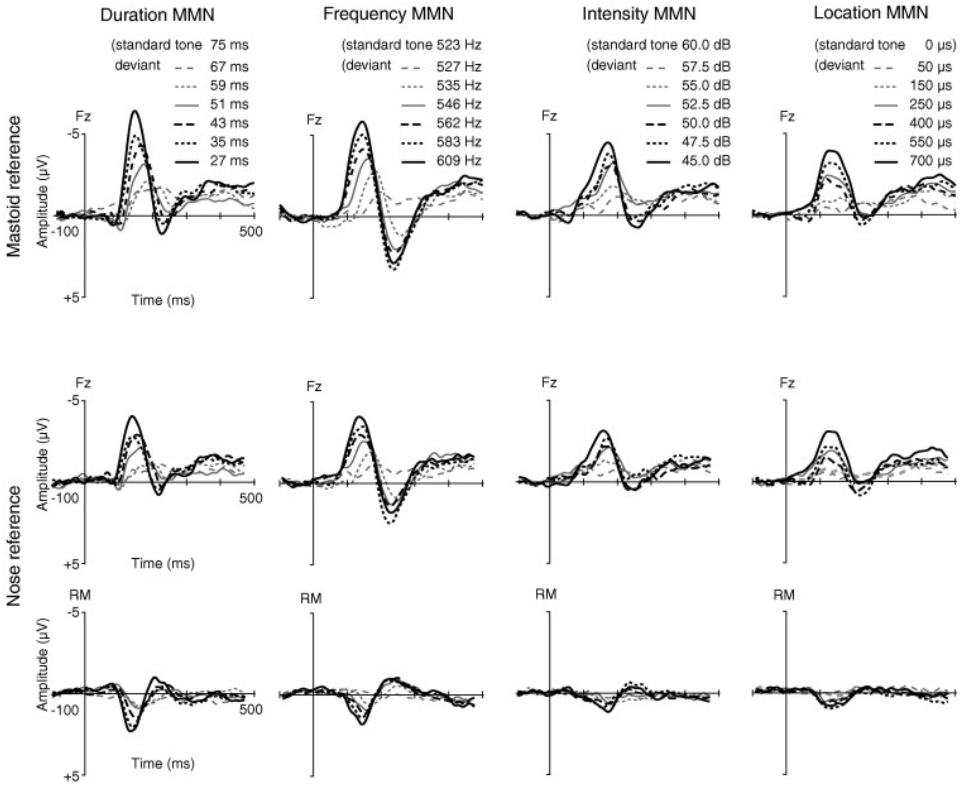

Figure 2.1: Example of resulting MMN waveforms for different types of oddballs (from [12])

by subtracting the ERP of the oddballs from the ERP of the regular stimulus. Fig. 2.1 shows examples of different MMN waveforms.

The effects of the oddball stimulus can be observed as early as 50 ms after the onset, with the peak often between 100-250 ms [9]. The first and largest peak is often a negative one, but the exact amplitude varies for different stimuli. The location where the negativity is most prevalent is at the frontocentral electrodes [9], with a tendency to be even more apparent in the right hemisphere.

The mismatch negativity is very useful to examine because there is a lot of available literature on it. This research also supports the notion that the MMN reflects cognitive processing for sound and speech [10–12] and that it can be used as a biomarker for certain disorders [2–4, 13], making it a useful tool for the KHL. The large amount of literature on this paradigm is also very useful as a means of validating the measurements and methods.

2.1.2. Acoustic Change Complex

Another way of eliciting an ERP that can be used to study sound processing by the brain is an ACC experiment. This effect is a bit newer than the MMN and was discovered by Ostroff, et al. [14] in 1998. In [14], the ACC effect was elicited by changing phonemes. Since its discovery more research was done with these phonemes, but also using other sound changes as well such as frequency, amplitude and source direction. Just like with the MMN, an ACC can be obtained even when no attention or response is given. It is also believed that the ACC reflects processing in the brain related to auditory discrimination or speech processing [15].

There are several relevant advantages of studying the ACC relevant for the KHL. First of all, the ACC has a higher signal-to-noise ratio (SNR) than the MMN [16]. This means it can be more clearly distinguished from other brain signals so that it can be analyzed more precisely. Another advantage is the reproducibility over different trials for the same person [17]. This makes it possible to track differences over time, that are not distorted by trial-to-trial variability. The third advantage for the KHL is that this

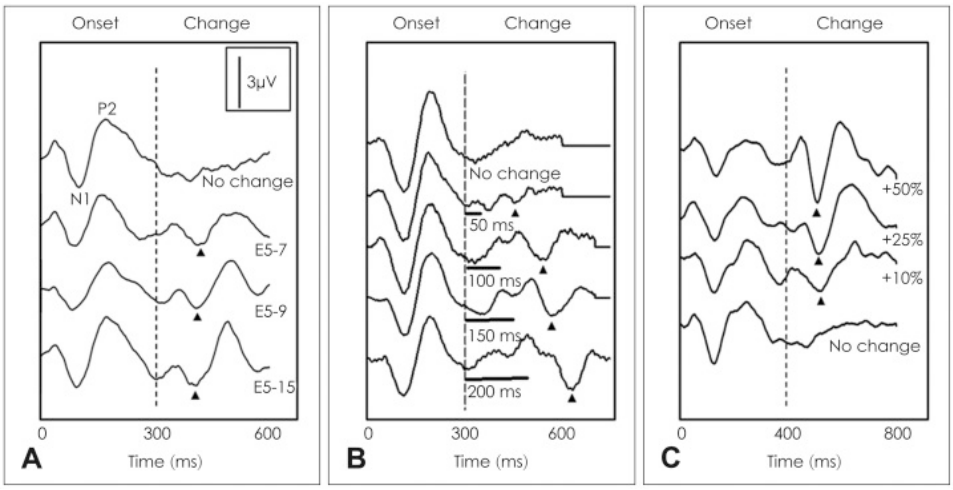

Figure 2.2: Example of several ACC waveforms induced by stimulating electrodes. A: Stimulus changes location. B: Stimulus has a pause. C: Stimulus increases in amplitude, starting amplitude is at 50% of the subjects dynamic range. (from [5])

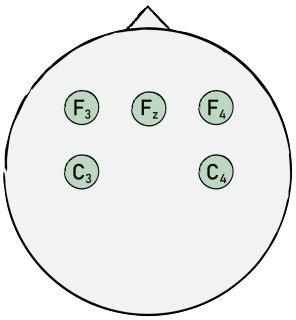

Figure 2.3: Locations of the electrodes used in the analysis.

effect can be seen in children [18, 19], who are, of course, the subjects of interest. Some examples of ACC waveforms are shown in Figure 2.2.

2.1.3. Measurements at Kinderhersenlab

The data used for this project are measurements of both MMN and ACC experiments done at the Kinderhersenlab. The measurements are done according to the *meetprotocol* (measurement protocol) [20]. The EEG recording is done with a geodesic sensor net (GSN), which is available in different sizes to accommodate different ages of subjects. The GSN used at the Kinderhersenlab has 128 electrodes or "channels", two of which are located at the mastoids and together serve as the reference electrode. This net can be applied very easily because no gel is needed to increase conductivity between the skin and the electrodes. Instead, the whole cap is submerged in a saline solution that is absorbed by sponges near the electrodes. Before recording, it is checked that the impedances of electrodes Fz, F3, C3, M1, M2, C4, F4 are below 50 $k\Omega$. Two of these electrodes, M1 and M2, are used as reference, and the others, of which the locations are shown in Fig. 2.3, are used in the analysis. All other electrodes should have an impedance below 100 $k\Omega$. The recording from the EEG cap is stored using the CURRY8 software. STIM2 is used to control the stimuli, this software sends the desired sound signals to the Nuevo System Unit and stores the timestamps in the CURRY8 file. The stimuli are presented according to the parameters that are set for each experiment.

Mismatch Negativity: To elicit the mismatch negativity, a regular stimulus and some oddball stimuli are used. These were presented by the aforementioned sound system according to the following parameters:

- The regular stimulus was a 1000 Hz tone, lasting 100 milliseconds,

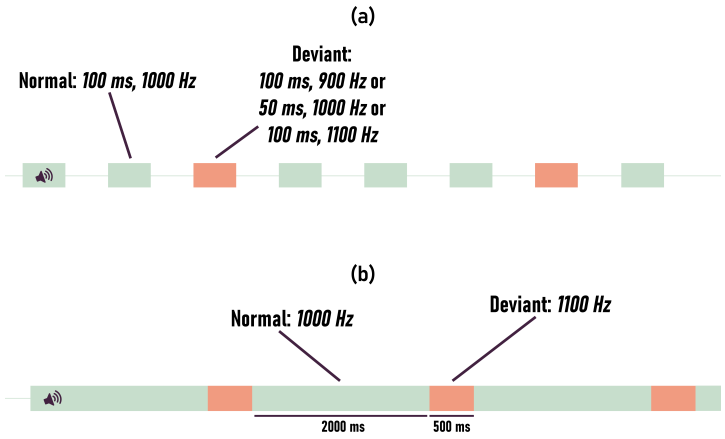

Figure 2.4: Visualization of the stimulus that are presented for (a) the MMN and (b) the ACC. The time axis is not to scale in the MMN image.

- The chance of a stimulus being regular was 80% of the time,
- The other stimuli (occurring with a probability of 20%) were one of three possible oddballs:
  1. the frequency of the tone was 900 Hz,
  2. the duration was 50 ms,
  3. or the frequency was 1100 Hz,
- All oddballs were equally likely to occur,
- A total of 790 tones were played during each trial.

An overview of both stimuli is given in Fig. 2.4.

Acoustic Change Complex: The acoustic change complex stimuli are also delivered using the same sound system. This experiment is executed a few minutes after the MMN and has the following parameters:

- A constant tone of 1000 Hz is presented,
- After 2000 ms, this tone switches to 1100 Hz,
- 500 ms after this frequency change, the frequency changes back to 1000 Hz,
- The above two steps are repeated 129 times.

2.1.4. Disorders Measured at KHL

The goal of the KHL is to measure children with a wide variety of disorders. Currently there are measurements available from subjects with seven different disorders:

- spina bifida (SB),
- GRIN/GRIA,
- Sturge-Weber syndrome (SWS),
- multiple sclerosis (MS),
- crouzon,
- persistent stuttering (PS),
- and brain overgrowth Syndrome (BO).

This section provides a short description of each disorder and the expected MMN and ACC effects from the literature.

Spina bifida is a birth defect that affects the vertebral column and often the spinal cord [21]. It often affects the cognitive abilities of children and adults suffering from this disorder, according to the literary review by Lindquist et al. [22]. This includes language processing suggesting the MMN and ACC can show differences for children with this disorder.

No specific studies relating SB with the MMN and ACC were found.

GRIN/GRIA is a group of genetic disorders linked to the GRIN genes responsible for establishing connections within the brain. It was first discovered in 2010 by Endele et. al. [23]. There are several different genes that can be affected in different ways to cause various symptoms [24], but the most common symptoms include developmental delays and delayed or no speech development. It might, therefore, have an effect on the MMN and ACC waveforms.

Crouzon is a syndrome which causes deformations of the skull, resulting in various other symptoms [25]. This generally does not decrease mental abilities, but it does in about 12% of cases [25]. In general it is therefore not expected for the disorder to have a significant effect on the ERP waveforms.

Sturge-Weber Syndrome is an innate disorder of blood vessels with various symptoms and a progression that is hard to predict [26]. The most apparent symptom is a naevus, often on one side of the face.

There are no MMN or ACC studies done for subjects with SWS. In general, there is no established relation between EEGs and neurodevelopment for SWS [27]. This is mainly due to the small number of patients, which results in low sample sizes.

Multiple sclerosis is a disease affecting the nervous system. This can lead to several symptoms, e.g. problems with vision and movement [28]. Around 40%-50% of MS patients also have a motor speech disorder, but it is poorly understood why and how this happens [29].

Jung et al. [30] have shown that the MMN effect is reduced in MS patients compared to controls. This reduction is stronger for patients with global cognitive impairment than for unimpaired patients. There are no conducted studies linking ACC effects to MS.

Persistent Stuttering is highly heterogeneous with regard to symptoms, avoidance behaviour, applied strategies to overcome disfluencies and severity. The symptoms of persistent stuttering have a high variety, as well as the causes. Key to fluent speaking is a feedback mechanism while speaking [31]. It is therefore not surprising that persistent stuttering causes reduced MMN effect [32]. In the research done by the EMC however, there is no effect of stuttering on the ERP.

Brain overgrowth syndromes is a group of syndromes that often have a genetic cause, some of these disorders are linked to neurological anomalies, such as cognitive impairment or autism [33]. Lopez-Arango et al. [34] show that children with brain overgrowth have different auditory event-related potentials. These differences are more apparent in a spectral analysis of the EEG but are also present in the ERPs. There is no literature linking brain overgrowth to MMN or ACC.

2.2. Signal Processing for the Electroencephalogram and Event-Related Potentials

In order to obtain the ERP signal of interest, some important preprocessing and signal processing steps have to be applied to the measurements. This is necessary because EEGs are always contaminated with certain types of noise and artefacts, and an ERP is never visible in a single recording. These processing steps to go from raw EEG data to useful ERP data are explained step-by-step in this section, and are summarized below:

- Apply a low-order band pass filter with cut-off frequencies of 0.1 and 30 Hz, as well as a notch filter with a centre frequency of 50 Hz.

- Reference all electrodes with regards to the specified referencing electrodes.
- Correct artefacts using independent component analysis. Components that contain artefacts are identified using the algorithm from Pion-Tonachini et al. [35].
- Divide the EEG into epochs that are time-locked by the stimulus onset.
- Remove epochs that contain artefacts that weren't corrected using several simple off-the-shelf artefact detection methods from EEGLAB [36] and ERPLAB [37].
- Average together all epochs from the same stimulus type to obtain the ERP waveforms.

2.2.1. Filtering

There are several types of noise that occur in EEG recordings that can be reduced using fairly simple filters. Low frequencies that are unrelated to brain activity often occur due to perspiration, which causes slow changes in impedance between the electrode and scalp. High frequencies, on the other hand, are caused by muscle movement from the face and neck. Both these types of noise can be reduced using a simple band-pass filter. Brain activity often happens in a few specific frequency bands, that are all between 1 and 30 Hz, so this would be appropriate cut-off frequencies for the filter. It is, however, important to note that filtering can distort the onset and offset times of ERP components, so this has to be kept in mind when analyzing results [7]. One preventative measure against these distortions is to not use filters with very harsh cut-offs. These hard transitions in the frequency domain, cause smoothing of the signal in the time domain. Another measure is setting the lower cut-off frequency lower than 1, e.g. 0.1 Hz.

Another noise source that can be removed using filters is line noise. This is 50 Hz noise (60 in the U.S. and some other countries), that is caused by the frequency of the power grid. This can be removed using a notch filter, tuned to 50 Hz.

2.2.2. Referencing

An EEG recording is a measure of potentials over the scalp, but a potential is no absolute value. A voltage is always in reference to some other point, often called the ground. The electronic circuit that is the EEG recorder is somewhat special in this regard, because it has both a ground and a reference point. The measured voltage at each channel is the voltage between the channel and the ground electrode. The voltage analyzed in most research is, however, the potential between a channel and one or more reference electrodes. This voltage of interest $\mathbf{x}_i'$ is obtained by simply subtracting away the reference voltage from each channel:

$$\mathbf{x}_i' = \mathbf{x}_i - \frac{1}{R}\sum_{r=1}^{R}\mathbf{x}_r, \tag{2.1}$$

where $\mathbf{x}_i$ is the original voltage over time on a single channel and $\mathbf{x}_r$ are the voltages over time over the R reference electrodes. Taking this approach to referencing allows the researcher to choose different reference points and cancels out noise from the ground circuit. Just like in a lot of other research, the KHL uses the average of two mastoid electrodes as a reference.

2.2.3. Artifact Correction

Next to noises, that are continuous random signals in certain frequency bands, an EEG is also distorted by artefacts. In contrast to noise, artefacts are often aperiodic and only last for brief periods of time [38]. To remove these unwanted signals, another strategy than filtering has to be applied. For EEG signals, the most apparent artefacts are caused by eye-blinking, but there are also other types of artefacts such as muscle artefacts, and cardiac activity [39]. When dealing with artefacts for ERP research, correction often refers to cleaning up the artifactual signals, while removal means discarding all ERP timeframes that contain an artefact.

Correcting artefacts in EEG signals is a well-researched topic. Lots of different algorithms have been developed in order to find the underlying signal of interest when an artefact occurs. Mumtaz et al. [40] lists the following possible processing categories; analogue methods, regression, adaptive filtering, in-

dependent component analysis (ICA), canonical correlation analysis (CCA), principal component analysis (PCA), wavelet transform (WT) decomposition, empirical mode decomposition (EMD) and hybrid methods. The hybrid methods combine two or more of the aforementioned techniques and most state-of-the-art techniques are among them. However, the most commonly used method is still ICA, as it can work very well and a lot of literature is available on it. Finding and implementing which state-of-the-art method works best for the data obtained in the KHL is outside the scope of this project, so ICA will be the method of choice.

Independent component analysis is a form of blind source separation (BSS). This means that it tries to find separate brain signals that are picked up by EEG electrodes over the skull without any prior knowledge of these signals. BSS can also be written as solving the following equation:

$$\mathbf{X} = \mathbf{AS}, \tag{2.2}$$

Where $\mathbf{X} \in \mathbb{R}^{N \times M}$ is a matrix containing the N measurements $\mathbf{x}_i \in \mathbb{R}^M$ from each electrode in the columns, $\mathbf{S} \in \mathbb{R}^{O \times M}$ are O independent signals that are picked up by the electrodes, and $\mathbf{A} \in \mathbb{R}^{N \times O}$ represents the relation between each signal and each electrode. This equation is almost always underdetermined and, therefore, has no analytical solution. Independent component analysis solves this problem by optimizing $\mathbf{A}$, such that the signals, or rows, in $\mathbf{S}$ are maximally statistically independent. This statistical independence or non-gaussianity can be measured using either kurtosis, skewness or negative entropy. The amount of independent components that are measured has to be predetermined, but is the same amount as the number of electrodes in the EEGLAB implementation.

Makeig et al. [41] suggest that some of these components contain artefacts, while others contain signals of interest. This is very useful because the original channels can be reconstructed from only the non-artifactual components. There are different algorithms to preprocess the EEG and to apply ICA, as well as different methods to select which components are artefacts and which are normal brain signals. More advanced methods such as combining ICA with complete ensemble empirical mode decomposition [42] or the wavelet transform [43] can in some cases outperform regular ICA.

In order to identify which ICA components are artifactual, a machine learning algorithm developed by Pion-Tonachini et al. [35] is used. This method is based on a large open-source EEG dataset with components being labelled by hand to train the ML algorithm.

2.2.4. Creating Epochs

As mentioned above, the ERP is often too small compared to other brain activity to be visible in a single trial. The common solution to this problem is to average over at least a few dozen trials. The start and end times of each trial are determined by the timestamp where the stimulus is presented. Often, this time of interest starts around 100-200 ms before the stimulus onset and lasts until about 600-1000 ms after the stimulus presentation. This period is called an epoch, and the EEG data is split up into these epochs before artefact removal.

2.2.5. Artifact Removal

By applying artifact correction methods the amount of artefacts in the recording can be reduced, but it is likely that some artefacts still remain present. To ensure the averaged ERP waveform is not distorted by this, epochs where this happened are thrown out. It is common practice to remove all channels of the epoch when at least one shows an artefact [7]. A few simple algorithms implemented as Matlab functions in ERPLAB [37] are applied in order to detect the artefacts in any of the channels.

Artefacts often differentiate from normal brain activity by their amplitude. Especially ocular artefacts can induce potentials far larger than relevant brain signals. These types of large amplitudes can, therefore, easily be removed by setting a simple threshold. This threshold has to be determined by visual inspection of the data because an appropriate one can vary a lot between measurement setups and recording sessions. If the data is similar enough over different recording sessions, a single threshold can be chosen for all the data from the same measurement setup. Thresholds for the other detection methods have to be determined this way as well.

One of these other methods is a slightly more sophisticated version of threshold detection. This algorithm does not set a threshold for an absolute value but for the difference between the highest and lowest value within the epoch. This works slightly better in some cases due to two reasons: some artefacts cause both a positive and negative peak, and it accounts for the normal brain activity being a bit higher or lower, which can dampen the artefact's absolute amplitude.

Combining these two fairly simple functions can remove most epochs that contain very apparent artefacts. Some smaller artefacts may go undetected, but those also distort the signal less and can be partially averaged out in the next step.

2.2.6. Averaging

As mentioned above, an individual event-related potential is almost never visible because other brain activity often has a larger amplitude. This other brain activity is, however, not time-locked to the epoch, while the ERP waveform is. This means that when the signal is averaged, the brain activity unrelated to the stimulus onset time cancels out.

Before averaging another step is necessary, which is baseline correction. The ERP waveform can have a superposition on the other brain processes. These other processes should cancel out as stated before, except for when there are trends spanning multiple epochs, such as the signal slowly drifting towards a higher or lower value. To correct for this, baseline correction takes the average signal amplitude before the stimulus onset and subtracts this from the entire epoch:

$$\mathbf{x}_i^{(p)} = \mathbf{x}_i^{(p)} - \frac{1}{T_{\text{pre-stimulus}}} \sum_{t=-T_{\text{pre-stimulus}}}^{0} x_i^{(p)}(t), \tag{2.3}$$

where $T_{\text{pre-stimulus}}$ is the number of samples in the epoch before the stimulus, and $\mathbf{x}_i^{(p)}$ is the signal over time for channel i and epoch p. After this step, the averaging over different epochs is quite straightforward:

$$\bar{\mathbf{x}}_i = \frac{1}{P} \sum_{p=1}^{I} \mathbf{x}_i^{(p)}, \tag{2.4}$$

where $\bar{\mathbf{x}}_i$ is a vector containing the average epoch for channel i, and $\mathbf{x}_i^{(1)}, \mathbf{x}_i^{(2)}, ..., \mathbf{x}_I^{(P)}$ are the individual epochs for channel I.

2.3. Event Related Potential Analysis

Ever since the ERP was discovered, researchers have been analyzing this signal. This research mainly focuses on studying brain processes, especially in the time domain, and cognitive disorders [44]. Traditionally, most ERP research on disorders was done by comparing the timing and amplitude of ERP components- [3, 45–50]. This has already provided vital information on these disorders, but components have more properties than just their delay and amplitude. These statistics do not provide enough precision to classify individual subjects and need relatively large sample groups to find these differences, so there is a need for better feature extraction and classification. In recent years, there have been many advancements in this field. Some of the commonly used methods that are relevant to this thesis are briefly outlined in this section.

2.3.1. Principal Component Analysis

Principal component analysis is a commonly used feature extraction method, often applied as a step before a machine learning classification algorithm. The idea behind PCA is to transform a matrix $\mathbf{X} \in \mathbb{R}^{I \times f}$, containing I samples with f features, such that the variance over the samples is maximized for as little features as possible. This is achieved by computing the covariance matrix of $\mathbf{X}$ and applying an eigenvalue decomposition. This results in both eigenvectors and eigenvalues. The eigenvectors are the new feature vectors that can be used for e.g. classification, and the eigenvalues are an indication of how much variance is explained by the corresponding eigenvector. By dividing each eigenvalue by the sum of all eigenvalues, the percentage of the total variance for each feature is obtained.

This is very useful for data with a huge amount of features or data points, such as EEG data, because a reduced amount of features can prevent overfitting. The use of PCA in feature extraction has, for instance, been shown in combination with a Bayesian classifier by Sun et al. [51] and k-nearest neighbour by Palaniappan et al. [52]. An overview of how PCA can be applied to ERP data is provided by Dien [53]. It is important to note that PCA is often applied as a first tool in feature extraction and classification, but very rarely used on its own.

2.3.2. Linear Discriminant Analysis

Just like PCA, linear discriminant analysis (LDA) is a commonly used method to extract features based on a transformation of the feature space. It is also called Fisher discrimination analysis (FDA) in some literature when only two classes are present. The transformation done by LDA can be used when all samples in the data belong to a certain class. LDA is also a bit like PCA in the sense that it tries to increase variance, but in this case not for all samples but between the different classes. The variance of all samples within a class is minimized on the other hand. This means in the new feature space, samples from the same class are more similar while the different classes are better distinguishable.

The way this works is by first calculating two scatter matrices, the between-scatter matrix $\mathbf{S}_b$ and the within-scatter matrix $\mathbf{S}_w$. They are calculated as:

$$\begin{aligned} \mathbf{S}_b &= \sum_{c=1}^{C} n_c(\bar{\mathbf{x}}_c - \bar{\mathbf{x}})(\bar{\mathbf{x}}_c - \bar{\mathbf{x}})^T, \\ \mathbf{S}_w &= \sum_{c=i}^{I} (\mathbf{x}_i - \bar{\mathbf{x}}_c)(\mathbf{x}_i - \bar{\mathbf{x}}_c)^T, \end{aligned} \tag{2.5}$$

where c is one of C different classes, n_c is the number of samples in class c, $\mathbf{x}_i$ is a vector containing the features of one sample, $\bar{\mathbf{x}}_c$ contains the average features of all samples within class c, and $\bar{\mathbf{x}}$ is a vector with the average features of all samples. The larger the differences between the class averages, the larger the values in $\mathbf{S}_b$ will be, and the larger the differences of samples within a class the larger the values in $\mathbf{S}_w$ will be. To better discriminate between classes, LDA aims to find a projection matrix that can be applied to the data $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_I]$. Just like in PCA, the projection matrix $\mathbf{U}$ is found using an eigenvalue decomposition. To deal with both scatter matrices instead of a single covariance matrix, the generalized eigenvalue decomposition (GEVD) is used:

$$\mathbf{S}_b\mathbf{U} = \mathbf{S}_w\mathbf{U}\mathbf{\Lambda}, \tag{2.6}$$

where $\mathbf{U} \in \mathbb{R}^{f \times f}$ contain the eigenvectors and $\mathbf{\Lambda}$ contains the corresponding eigenvalues. To reduce the number of features, the matrix $\mathbf{U}$ can be reduced by only keeping the eigenvectors corresponding to the R largest eigenvalues, resulting in $\mathbf{U} \in \mathbb{R}^{f \times R}$.

Effectively applying LDA on EEG and ERP data has been demonstrated [54, 55]. The downside is, however, that LDA is ill-suited for datasets containing many features and few samples. This is why LDA is often used in combination with other feature reduction or selection techniques in these cases [56], such as PCA [51]. Another solution to the small dataset problem is adding a regularization term, which was used on EEG data by Peterson et al. [57].

2.3.3. K-Nearest Neighbours

A topic that has gained huge attention in many fields of research is machine learning. This is also the case for EEG and ERP studies [58]. Especially the simpler ML models are suitable for EEG research, because of the limitations in sample sizes. The model most relevant to this thesis project is described below.

The k-nearest neighbours (KNN) algorithm is fairly simple. When a new sample has to be classified, the k nearest samples from the training data are identified. The new sample is then classified as being the same class as the majority of the k neighbours. In higher dimensional spaces, there are several methods to define the distance between points and, therefore, which samples are closest to each other. The most commonly used is the Euclidean distance.

KNN has also been applied for EEG classification [59–63]. The main advantages of KNN in this field of study are its suitability for low sample sizes and low computational complexity.

2.3.4. Regression

Machine learning cannot only be used for classification but also for regression when the desired output is not a class but instead a continuous variable. This means the output of the model will be an approximated outcome based on certain inputs. There are many different types of regression models. Most of them are based on finding parameters for a specified set of functions that provide a good fit for the training data. This is often done by minimizing a cost function, that can for instance be mean square error (MSE):

$$\min_{\mathbf{w}} \frac{1}{N}||\mathbf{y} - f(\mathbf{X}, \mathbf{w})||_2^2, \tag{2.7}$$

where N is the amount of training samples, $\mathbf{y}$ a vector containing the outputs of the training data, $\mathbf{X}$ contains multiple inputs for each training sample and $\mathbf{w}$ are the weights of function $f(\mathbf{x})$. This is just one example of how a regression model can be made, but there is an infinite amount of variations on this by using different loss functions, constraints and functions. Often a regularization term is added to the minimization problem to prevent overfitting the function to the training data.

2.3.5. Tensors

A growing field of research is that of tensors and tensor networks. Tensors are data structures, just like scalars, vectors and matrices, but of a higher order. So where a matrix is always of size $A \times B$, a tensor can have a size $I_1 \times I_2 \times ... \times I_N$, where I_n is called a mode and the order is N. Many types of data are naturally occurring in this form, such as video (*vertical pixels* $\times$ *horizontal pixels* $\times$ *timeframes*) or library indexing (*floor* $\times$ *bookcase* $\times$ *shelf* $\times$ *book position*). Most methods of analyzing these types of data are, however, based on vectors and matrices. To still be able to use feature extraction and machine learning, the higher order data is often "flattened" into vector form, but by doing this valuable information on the relation between variables can get lost. Dealing with tensors directly does not have this disadvantage, so it is becoming increasingly popular.

In the study of EEGs and ERPs, tensors are becoming more common as well [64]. An ERP can be written in tensor form in various ways, e.g. Zhao et al. [65] uses a tensor with modes *channel* $\times$ *time* $\times$ *frequency* $\times$ *subject*. Bonab et al. [66] use a tensor with modes *channel* $\times$ *time* $\times$ *trial* in order to denoise ERP waveforms. The ability to deal with tensors directly, instead of vectorizing, in these and most other tensor studies comes from some uncommon mathematics that are used. In addition, tensor decompositions are also often used when dealing with this data type. A brief overview of tensor notation and mathematics is therefore provided below. A short introduction to two of the most common tensor decompositions can also be found in this section.

Tensor Notation and Mathematics: The following notations are used to distinguish between different types of arrays in this report:

- scalars are denoted as lowercase letters: x,
- vectors are denoted as bold lowercase letters: $\mathbf{x}$,
- Matrices are denoted as bold uppercase letters: $\mathbf{X}$,
- Tensors are denoted as underlined bold uppercase letters: $\underline{\mathbf{X}}$.

The following operations for tensors $\underline{\mathbf{A}}, \underline{\mathbf{C}} \in \mathbb{R}^{I_1 \times I_2 \times ... \times I_N}$ and $\mathbf{B} \in \mathbb{R}^{I_k \times I_l}$ are commonly used:

- *vec*$(\underline{\mathbf{A}})$ vectorizes a tensor: $\underline{\mathbf{A}} \in \mathbb{R}^{I_1 \times I_2 \times ... \times I_N} \rightarrow \underline{\mathbf{A}} \in \mathbb{R}^{1 \times I_1 I_2 ... I_N}$,
- $\mathbf{A}_{(k)}$ denotes the mode-k matricization: $\underline{\mathbf{A}} \in \mathbb{R}^{I_1 \times I_2 \times ... \times I_N} \rightarrow \mathbf{A}_{(k)} \in \mathbb{R}^{I_k \times I_1 ... I_{k-1} I_{k+1} ... I_N}$,
- $\underline{\mathbf{A}} \times_k \mathbf{B}$ is the mode-k product: $(\underline{\mathbf{A}} \times_k \mathbf{B})_{i_1 ... i_{k-1} i_l i_{k+1} ... i_N} = \sum_{i_k}^{I_l} a_{i_1 i_2 ... i_N} b_{i_k i_l}$, resulting in $\underline{\mathbf{Y}} \in \mathbb{R}^{I_1 \times ... \times I_{k-1} \times I_l \times I_{k+1} \times ... \times I_N}$.
- $\underline{\mathbf{A}} \times_k^l \mathbf{B}$ is the mode-k, l-product: $(\underline{\mathbf{A}} \times_k^l \mathbf{B})_{i_1,...,i_{k-1},i_{k+1},...,i_{l-1},i_{l+1},...,i_N}$ $= \sum_{i_k=1}^{I_k} \sum_{i_l=1}^{I_l} \underline{\mathbf{A}}_{i_1,...,i_k,...,i_l,...,i_N} \cdot b_{i_k,i_l}$, resulting in $\underline{\mathbf{Y}} \in \mathbb{R}^{I_1 \times ... \times I_{k-1} \times I_{k+1} \times ... \times I_{l-1} \times I_{l+1} \times I_N}$.

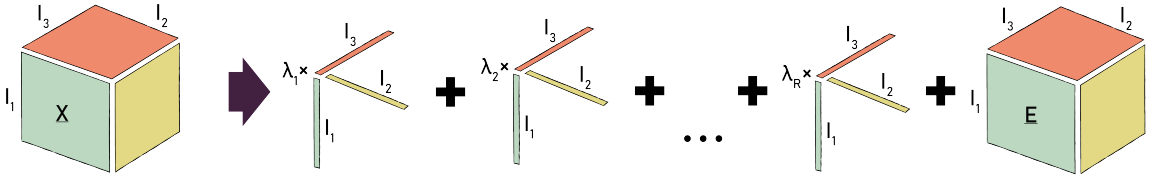

Figure 2.5: Visualization of CPD decomposition of tensor with three modes.

- $||\underline{\mathbf{A}}||_F$ is the Frobenius norm: $||\underline{\mathbf{A}}||_F^2 = \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} ... \sum_{i_N=1}^{I_N} a_{i_1,i_2,...,i_N}^2$
- $\langle \underline{\mathbf{A}}, \underline{\mathbf{C}} \rangle$ is the inner product: $\langle \underline{\mathbf{A}}, \underline{\mathbf{C}} \rangle = \sum_{i_2=1}^{I_2} ... \sum_{i_N=1}^{I_N} a_{i_1,i_2,...,i_N} c_{i_1,i_2,...,i_N}$

Tensor decompositions: Two basic and very commonly used tensor decompositions are the canonical polyadic decomposition (CPD), which is sometimes called PARAFAC or CANDECOMP, and the Tucker decomposition (TD).

Canonical Polyadic Decomposition: The CPD is a very useful tool in signal processing, because it can decompose a tensor into rank R outer products of vectors:

$$\underline{\mathbf{X}} = \sum_{r=1}^{R} \lambda_r \mathbf{b}_r^{(1)} \circ \mathbf{b}_r^{(2)} \circ ... \circ \mathbf{b}_r^{(N)} + \underline{\mathbf{E}}. \tag{2.8}$$

In this equation, N denotes the order of $\underline{\mathbf{X}}$, λ_r are scalars, and $\underline{\mathbf{E}}$ is an error term. This error is added because not all tensors can be perfectly decomposed as a rank R decomposition. A visualization of the CPD written as vector outer products is shown in Figure 2.5. The CPD can also be written in terms of a super diagonal tensor $\underline{\mathbf{\Lambda}}$, containing the scalars λ_r, and matrices:

$$\underline{\mathbf{X}} = \underline{\mathbf{\Lambda}} \times_1 \mathbf{B}^{(1)} \times_2 ... \times_N \mathbf{B}^{(N)} + \underline{\mathbf{E}}. \tag{2.9}$$

The CPD can also be denoted as:

$$\underline{\mathbf{X}} = [\![\underline{\mathbf{\Lambda}}; \mathbf{B}^{(1)}, \mathbf{B}^{(2)}, ..., \mathbf{B}^{(N)}]\!]. \tag{2.10}$$

The CPD is essentially unique under mild conditions [67]. Applying CPD is often used to extract useful features from a tensor, which can be applied to ERP data as well [64, 68].

Tucker Decomposition: The Tucker decomposition is a bit similar to the CPD, it can also be written as Eq. 2.9, but the superdiagonal tensor $\underline{\mathbf{\Lambda}}$ is replaced by a core tensor $\underline{\mathbf{G}}$:

$$\underline{\mathbf{X}} = \underline{\mathbf{G}} \times_1 \mathbf{B}^{(1)} \times_2 ... \times_N \mathbf{B}^{(N)}. \tag{2.11}$$

The matrices $\mathbf{B}^{(n)}$ are called the factor matrices. If $\underline{\mathbf{X}} \in \mathbb{R}^{I_1 \times I_2 \times ... \times I_N}$, then $\underline{\mathbf{G}} \in \mathbb{R}^{R_1 \times R_2 \times ... \times R_N}$ and $\mathbf{B}^{(1)} \in \mathbb{R}^{R_1 \times I_1}, \mathbf{B}^{(1)} \in \mathbb{R}^{R_2 \times I_2}, ..., \mathbf{B}^{(N)} \in \mathbb{R}^{R_N \times I_N}$. The sizes of modes $R_1, R_2, ..., R_N$ are called the ranks of the decomposition and can have different values. For the TD there is no constraint on the core tensor, meaning the uniqueness from the CPD is not the case for the TD. A visualization of the Tucker decomposition is shown in Figure 2.6.

TDs are used for studying EEGs as well. Zhao et al. [65] use a variant of the TD called multilinear singular value decomposition. In [64] the TD is compared to the CPD with applications in EEG signal processing.

2.3.6. Discussion

There are many different ways in order to analyze EEG and ERP data. The classical way of using ERP components peak and latency can show there is a difference in cognitive processing for some disorders. This is, however, only visible when looking at groups, and not often for individual subjects. By selecting

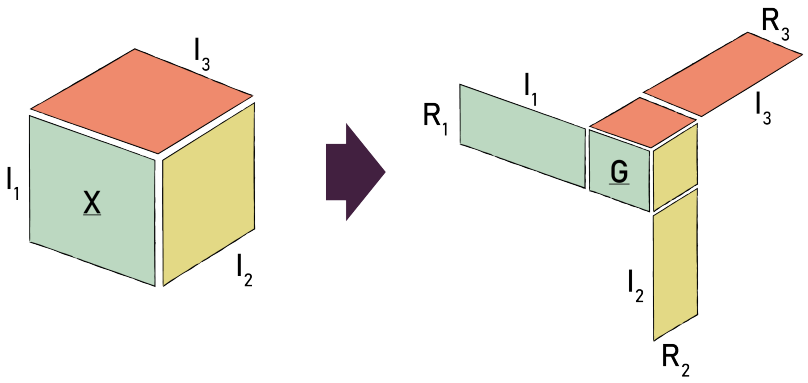

Figure 2.6: Visualization of Tucker decomposition for tensor with 3 modes.

and extracting features using linear algebra methods, such as PCA, LDA, and ICA, classification of specific brain activity with reasonable accuracy becomes possible for individual patients [69–73]. The classification using these features is often performed by machine learning models. These methods do, however, often not take into account the naturally forming tensor data structure of ERP measurements. Tensor networks have also gained popularity for analyzing EEG and ERP signals and have proved very useful for both analysis and classification.

For this thesis, the tensor data structure of ERP measurements will be further explored. Since LDA is a very useful tool for classification, but also to find features on which the classification is based, higher order extensions to this algorithm can be very well suited for EEG data. There is already literature available on this [74–78], with some studies specifically for EEG signals [79, 80]. These tools can prove very useful for classifying the different disorders measured at the EMC, as well as finding the features that distinguish the disorders. In order to find relations between the ERP signals and cognitive development, regression can be used. Although originally limited to vectors and matrices, methods for tensor regression have also been developed [81–85]. Both higher-order discriminant analysis and tensor regression will be further explained in the methodology chapter, along with how they are applied to the KHL measurements.

3

ERP Analysis

Before moving on to more advanced techniques to analyse the measurements from the child brain facility, it is wise to first study the ERP waveforms. These waveforms are generated by the method described in Section 2.2. In this chapter, a short description of the data received by the KHL will be given. Then the ERP plots coming from this data will be shown and discussed in the subsequent sections of this chapter.

3.1. Data from KHL

In total, 54 subjects are measured at the KHL at the time of the analysis for this thesis. For some of these subjects, one of the two tests was cut short or cancelled altogether because sitting still and listening to a harsh sound was not easy for some children. In the processing pipeline, a selection of "bad" epochs was made. Subjects that had more than 40% of their epochs removed are excluded for further analysis in order to prevent low-quality ERPs from polluting the average waveform. This results in 31 subjects having produced a successful MMN measurement and 14 subjects for the ACC. The number of successful ACC measurements is lower, which is likely due to the constant ACC tone is more unpleasant for the children to listen to, resulting in more head movements and shorter trials.

3.2. Mismatch Negativity Waveforms

In this section, the average of all MMN waveforms, called the grand average (GA), will be discussed, as well as the average per disorder.

3.2.1. Grand Average

The resulting ERPs from the MMN experiment can be seen in two figures. Fig. 3.1 shows the waveform produced by the regular stimulus, as well as the waveforms produced by the deviants. The figures indicate that the measurements were successful because they show similarities to what is expected from the literature. Two large peaks are the main interesting thing, a positive one around 80 ms and a negative one around 250 ms. these are visible for both the regular and deviant stimuli. The two frequency deviants look very similar, which is also something that would be expected. The duration of the deviant waveform deviated from the original waveform a bit later, which is due to the fact that it takes 50 ms longer for the brain to register this stimulus as different.

From the four different ERP waveforms, the three MMN waveforms can be made. These are the differences between the regular ERP and the deviants. The resulting difference waveforms can be seen in Fig. 3.2.

3.2.2. Average per Disorder

The grand average of all subjects indicated that the measurements at the KHL were successful, so in this section, the measurements are used to see if any conclusions can be drawn about the differences between the various disorders. For surveyability, this analysis is done at the Fz electrode because this

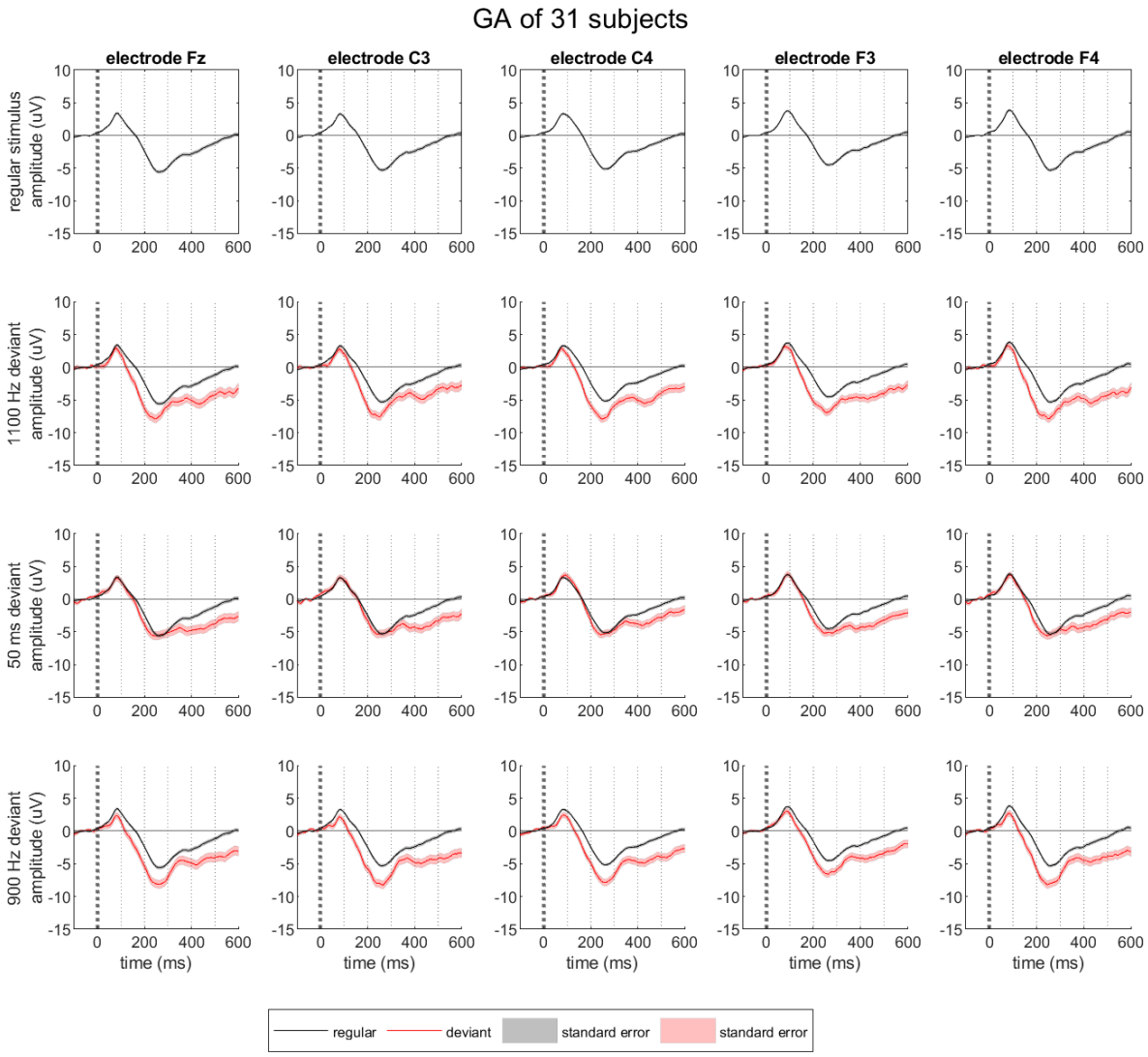

Figure 3.1: Grand average ERP of 31 subjects that resulted in successful MMN measurement. The columns indicate measurements from different electrodes, and the rows show either the regular stimulus (1000 Hz, 100 ms) or the regular stimulus with a deviant that has either a different tone or duration.

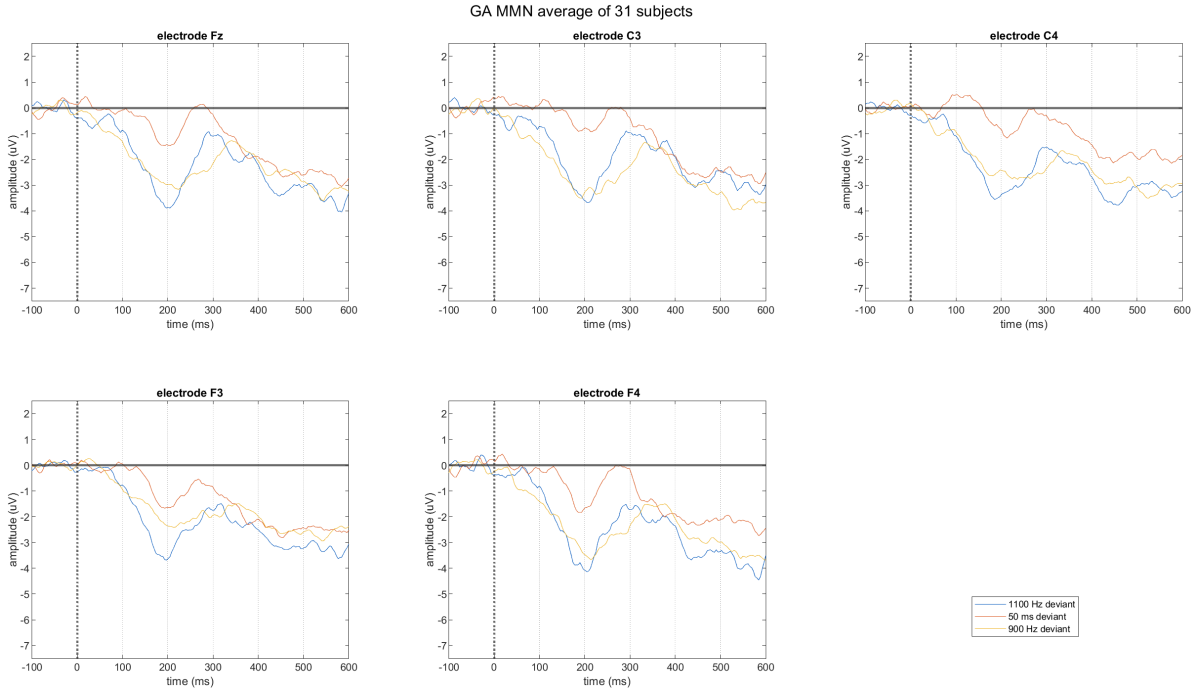

Figure 3.2: Mismatch negativity waveforms at different electrodes. These difference waveforms are made by subtracting a deviant ERP from the regular stimulus (1000 Hz, 50 ms).

is the centrally placed electrode, and no big differences between electrodes were seen in the GA.

In Fig. 3.3 an overview is given of the regular and deviant ERPs, averaged for each disorder. By looking at this overview, a few conclusions can be made before looking at the different waveforms.

- Crouzon, MS and brain overgrowth syndrome have a low number of subjects. This could mean that abnormalities in the waveforms of these disorders can be caused by a single subject, making it not representative of the disorder in general.
- In general, there are a lot of places where the standard errors of the regular and deviant stimuli overlap. At these places, drawing conclusions from the difference waveforms is impossible.

With these considerations in mind, a look can be taken at Fig. 3.4. This figure shows a clearer overview of the different disorders in the form of the MMN difference waveforms, along with the grand average. For Sturge-Weber, crouzon and stuttering, no effect of the disorder on the MMN waveform are expected based on the literature. Comparing these disorders with each other and the grand average, this appears to be true for the KHL measurements as well. GRIN/GRIA is also fairly similar to these disorders, but the negative component at 200 ms has a higher amplitude than in most other waveforms. the brain overgrowth average also stands out from the other ERPs, but these differences cannot be reliably attributed to the disorder itself due to the low number of subjects.

GRIN/GRIA and Sturge Weber are both disorders with a relatively large number of subjects for the MMN measurements. This results in a waveform in which the N1 component is clearly distinguishable, making it possible to compare amplitudes and delays. In Fig. 3.5 the amplitude and delay of the average waveforms for both these disorders are shown. Because there is a bit of skewness in both components, the delays of the peaks are not representative of the whole component. It is, therefore, more interesting to look at Fig. 3.6; in this figure, the delay and amplitude of the “centre of gravity” of the component are shown.

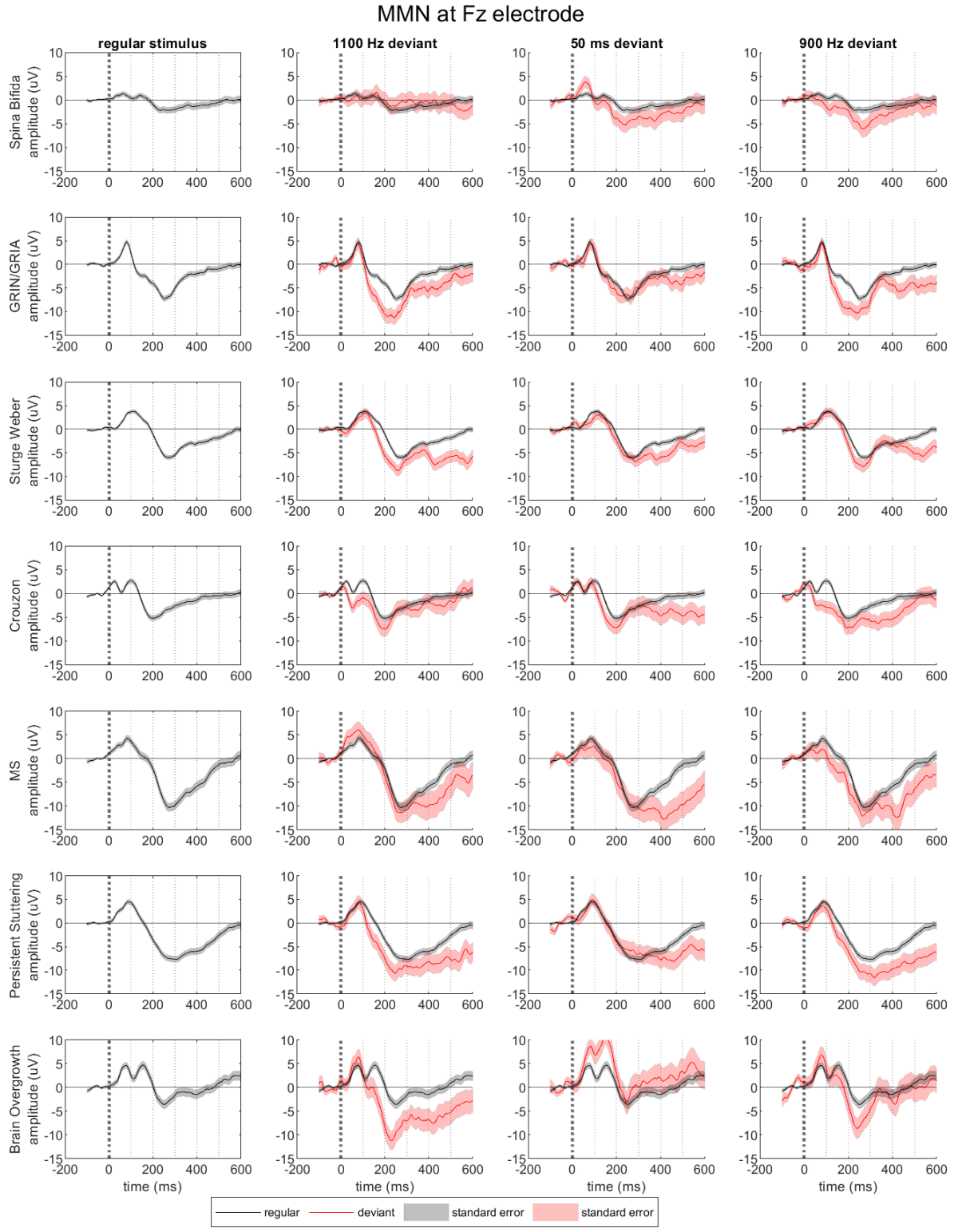

Figure 3.3: Overview of regular ERPs and deviant ERPs, averaged for each disorder. The averages were constructed using 7 subjects for spina bifida, 8 for GRIN/GRIA, 8 for Sturge Weber, 3 for crouzon, 2 for MS, 4 for Persistent stuttering and 2 for brain overgrowth syndrome.

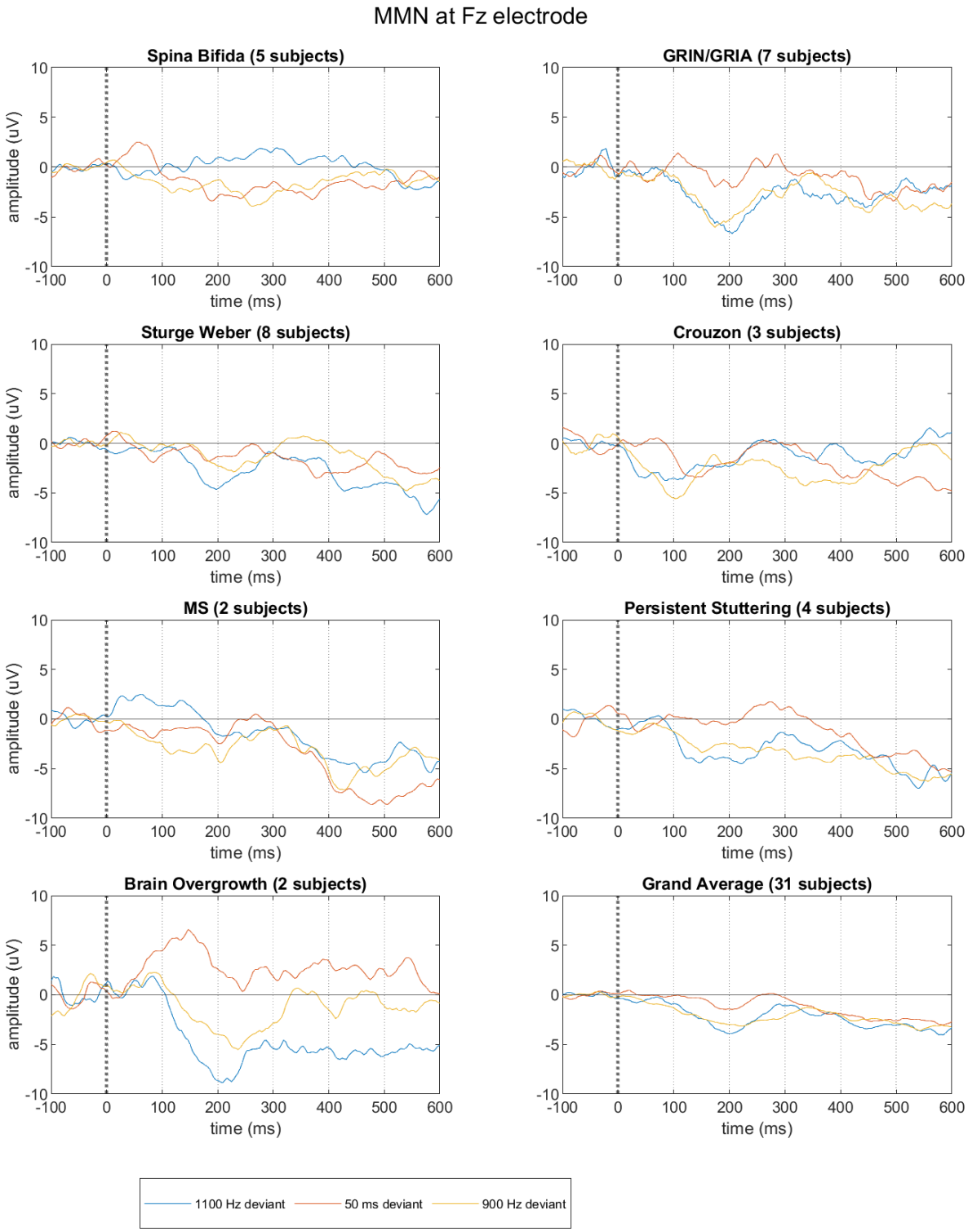

Figure 3.4: Average MMN waveform for each disorder and the GA.

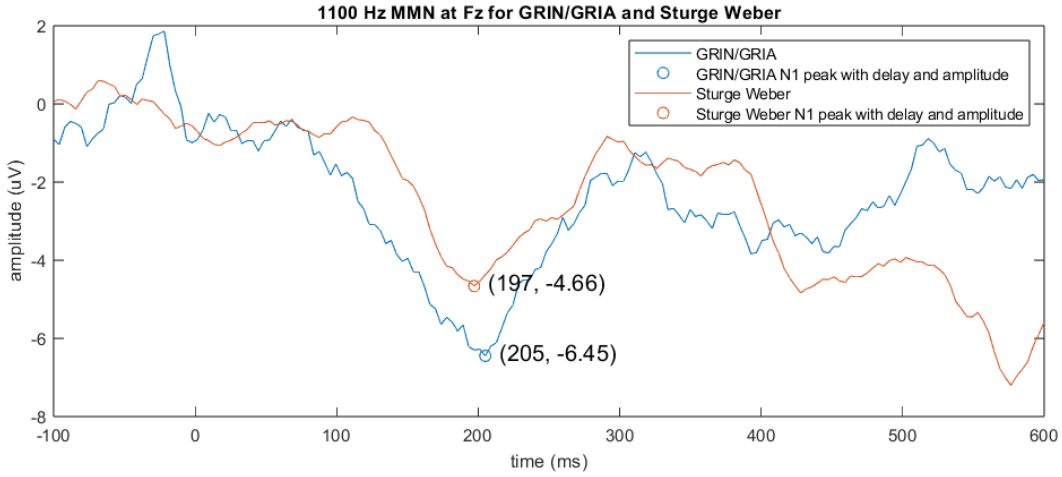

Figure 3.5: average MMN waveforms at Fz for subjects with GRIN/GRIA and Sturge Weber. The amplitudes and delays of the peaks of the N1 component are noted in the figure.

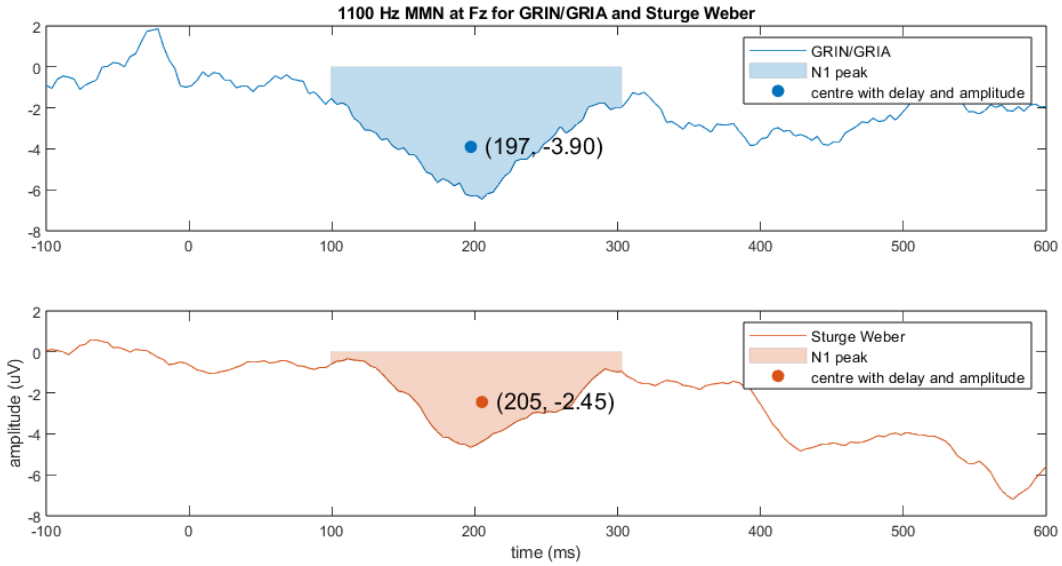

Figure 3.6: average MMN waveforms at Fz for subjects with GRIN/GRIA and Sturge Weber. The average amplitudes and delay of the shaded area that corresponds to the N1 component are noted.

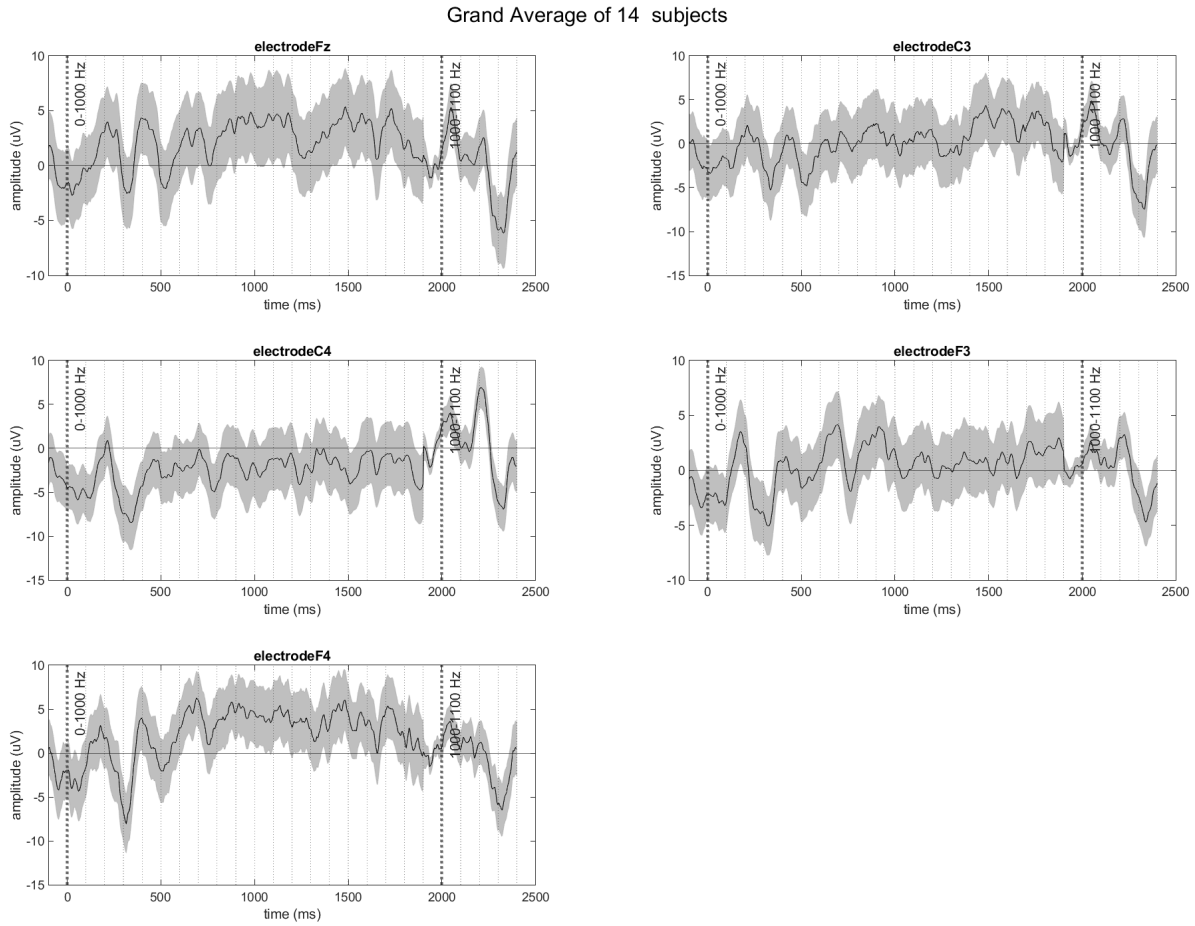

Figure 3.7: Grand average waveform ACC waveform of 11 subjects, shown for five electrodes.

3.3. Acoustic Change Complex Waveforms

In this section, the grand average waveform of the ACC ERPs is discussed first. The averages of the various disorders are shown as well and compared with each other and the GA.

3.3.1. Grand Average

The grand average of all successful ACC trials can be seen in Fig. 3.7. The baseline correction before averaging is done using the signal between 1900-2000 ms because the waveform before 0 seconds still shows some effects from the acoustic change at 2000 ms. The quality of this GA is a bit lower than that of the MMN due to the lower number of subjects used in the averaging. Nevertheless, a clear ACC effect can be seen after 2000 ms, when the frequency changes from 1000 to 1100 Hz. The effect at 0 ms is also apparent in some electrodes, although it is less clear. This is not surprising because at $t = 0$, the change from 1000 to 1100 Hz happened only 500 ms ago, and the longer the inter-stimulus time, the larger the ACC effect is [86].

3.3.2. Average per Disorder

Looking at the ERPs of the different disorders in Fig. 3.8, there are some clear differences between the disorders. These differences are, however, not that reliable due to the low number of subjects per disorder. For brain overgrowth syndromes, there are no subjects that have a successful measurement.

Looking at the disorders where no difference from healthy subjects is expected, they are not as similar as for the MMN measurements. Sturge-Weber syndromes ERP and Persistent stutterings ERP both have a large negative-going peak around 2200 ms stimulus, but crouzons shows no effect. It is interesting to note that GRIN/GRIAs average shows a large positive going peak before going down again.

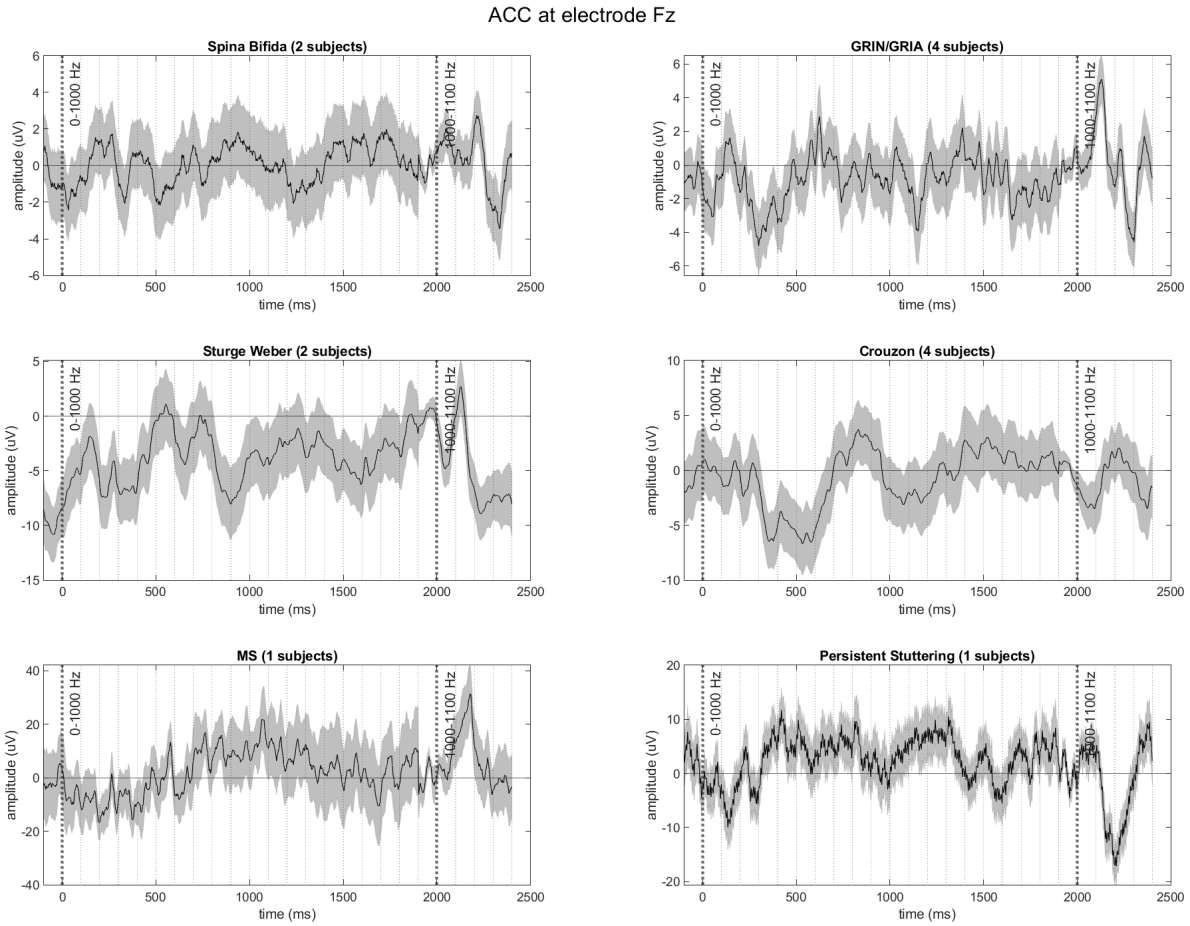

Figure 3.8: Average ACC waveform for subjects with the same disorder, measured at the Fz electrode.

Nevertheless no reliable conclusions can be made from these ERPs due to the low number of subjects per disorder.

3.4. Relation MMN and ACC Waveforms

When comparing the ERPs from both experiments that are performed at the KHL with each other, some careful conclusions can be made. Most disorders show no significant differences from each other; in MMN, this is due to the ERPs being quite similar, and for ACC, this is due to the low number of subjects and, therefore, unreliable average. In ERPs for both experiments, the disorders that should have no effect are quite similar to each other, although this is not the case for the ACC average of crouzon. The most interesting observation is the large amplitude in GRIN/GRIA, which happens around 200 ms in the MMN ERP and around 2100 ms in the ACC ERP.

3.5. Discussion

The grand averages of both the MMN and ACC waveforms show that the measurements results in some ERP components that are expected from literature. When looking at the differences between the various disorders however, no clear conclusion can be made especially for the ACC averages. The one disorder that stands out compared to the others in both the MMN and ACC waveforms is GRIN/GRIA. More interesting observations can hopefully be made by using more advanced analysis techniques described in the next chapter.

4

Methodology

Both discriminant analysis and regression algorithms are developed to deal with samples with their features described in a vector. When data naturally occurs as a tensor, using these methods requires the data to be flattened to a vector. Because valuable information on the relations between variables gets lost by this operation, this thesis aims to explore methods for classification and regression that can directly operate on a tensor. The algorithms used in this project can be divided into two categories: tensor discriminant analysis and tensor regression. Tensor discriminant analysis is used to find linear transformations that make it easier to distinguish the different disorders. The accuracy of classifying the disorders in individual subjects can be studied for these projections. Alternatively, regression models should be used when the problem at hand does not involve binary classification or categorical variables but instead involves a continuous outcome variable. Such a model aims to estimate one or more output variables based on a (tensor) input.

This chapter first discusses discriminant analysis, starting with a simple algorithm. Alternatives and extensions to this algorithm are also described in the same section. For tensor regression, a basic model is also given, which is also extended using assumptions about the KHL data.

4.1. Discriminant Analysis for Tensors

In Chapter 2, linear discriminant analysis is already introduced. This method aims to find a projection matrix that can be applied to a matrix containing subjects from different classes with some features to better distinguish them in a lower dimensional space. For LDA, these features have to be in the form of vectors, which, stacked together, form a matrix. Different algorithms have to be used when the data is given in tensor form, as is the case for the KHL data. This chapter describes a fairly basic extension of LDA to tensors called higher-order discriminant analysis (HODA). This algorithm results in several projection matrices, and by applying constraints to these projection matrices, two alternative algorithms are described as well.

In this thesis, the classes $c \in \{1, 2, \ldots, C\}$ that have to be predicted are the various disorders of the subjects. The modes of the input tensors vary a bit depending on the experiment and possible alterations of the input, but are in general time, electrodes and experiment.

4.1.1. Higher Order Discriminant Analysis

Discriminant analysis for higher-order data was first developed by Yan et al. [74], and this is a form of multilinear discriminant analysis (MLDA). In their research, they coined the term discriminant analysis with tensor representation (DATER), but in later research, the term higher order discriminant analysis is used more often. They proposed a method to find a series of projection matrices $\mathbf{U}_1 \in \mathbb{R}^{I_1 \times I_1'}, \mathbf{U}_2 \in \mathbb{R}^{I_2 \times I_2'}, ..., \mathbf{U}_K \in \mathbb{R}^{I_K \times I_K'}$ that can reduce the size of a tensor containing the features of a subject $\underline{\mathbf{X}} \in \mathbb{R}^{I_1 \times \ldots \times I_K}$:

$$\underline{\mathbf{X}}' = \underline{\mathbf{X}} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times_3 ... \times_K \mathbf{U}_K, \tag{4.1}$$

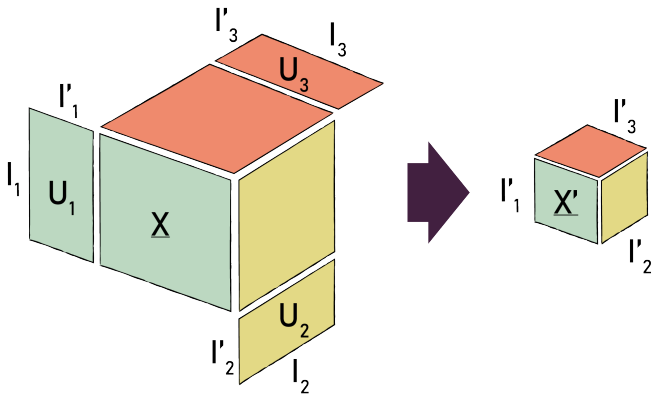

Figure 4.1: Visualization of feature reduction using higher order discriminant analysis on a third order tensor.

where $\underline{\mathbf{X}}' \in \mathbb{R}^{I'_1 \times \ldots \times I'_K}$ is a mapping of the features that contain the most valuable information for classification. This operation is visualized in Fig. 4.1.

The optimization problem that is solved for $\mathbf{U}_1, \mathbf{U}_2, ..., \mathbf{U}_K$ to find these factor matrices is the following:

$$(\mathbf{U}_k^*|_{k=1}^K) = arg \max_{\mathbf{U}_k|_{k=1}^K} \frac{\sum_c^C n_c ||\bar{\underline{\mathbf{X}}}_c \times_1 \mathbf{U}_1 \times_2 \ldots \times_K \mathbf{U}_K - \bar{\underline{\mathbf{X}}} \times \mathbf{U}_1 \times_2 \ldots \times_K \mathbf{U}_K||^2}{\sum_i^I ||\mathbf{X}_i \times_1 \mathbf{U}_1 \times_2 \ldots \times_K \mathbf{U}_K - \underline{\mathbf{X}}_{c_i} \times_1 \mathbf{U}_1 \times_2 \ldots \times_K \mathbf{U}_K||^2}. \tag{4.2}$$

This problem is solved for I subjects in C different classes. $\underline{\mathbf{X}}_i$ is a tensor for a single subject, $\bar{\underline{\mathbf{X}}}_c$ is a tensor containing the features averaged for all subjects in class c, $\underline{\mathbf{X}}_{c_i}$ is the class average of the class to which subject i belongs, and $\bar{\underline{\mathbf{X}}}$ contains the average features of all subjects. Eq. (4.2) cannot be solved analytically for all $\mathbf{U}_k$ at once, but it can be solved by iterating over each $\mathbf{U}_k$. At every iteration, this is done by using the between scatter matrix $\mathbf{S}_b$ and within scatter matrix $\mathbf{S}_w$:

$$\begin{aligned}
\underline{\mathbf{Y}}_{(k),i} &= \underline{\mathbf{X}}_i \times_1 \mathbf{U}_1 \times_2 \ldots \times_{k-1} \mathbf{U}_{k-1} \times_{k+1} \mathbf{U}_{k+1} \times_{k+2} \ldots \times_K \mathbf{U}_K, \\
\mathbf{S}_b &= \sum_{j=1}^{\prod_{o \neq k} I_o} \mathbf{S}_b^j, \quad \mathbf{S}_b^j = \sum_{c=1}^{C} n_c (\bar{\mathbf{Y}}_{(k),c}^j - \bar{\mathbf{Y}}_{(k)}^j)(\bar{\mathbf{Y}}_{(k),c}^j - \bar{\mathbf{Y}}_{(k)}^j)^T, \\
\mathbf{S}_w &= \sum_{j=1}^{\prod_{o \neq k} I_o} \mathbf{S}_w^j, \quad \mathbf{S}_w^j = \sum_{i=1}^{I} (\mathbf{Y}_{(k),i}^j - \bar{\mathbf{Y}}_{(k),c_i}^j)(\mathbf{Y}_{(k),i}^j - \bar{\mathbf{Y}}_{(k),c_i}^j)^T,
\end{aligned} \tag{4.3}$$

where $\bar{\mathbf{Y}}_{(k),c_i}$ and $\bar{\mathbf{Y}}_{(k),c}$ are the features averaged over each subject in a certain class, $\bar{\mathbf{Y}}_{(k)}$ contains the average features over all subjects and j indicates a column slice, as illustrated in Fig. 4.2. $\underline{\mathbf{Y}}_{(k),i}$ is similar to $\underline{\mathbf{X}}'$, but the projection matrix is not applied to mode-k.

When the scatter matrices have been found, they can be used to find $\mathbf{U}_k$ by solving:

$$\underset{\mathbf{U}_k}{\arg\max} \frac{\mathbf{U}_k^T \mathbf{S}_b \mathbf{U}_k}{\mathbf{U}_k^T \mathbf{S}_w \mathbf{U}_k}, \tag{4.4}$$

which can be done using the generalized eigenvalue decomposition (GEVD):

$$\mathbf{S}_b \mathbf{U}_k = \mathbf{S}_w \mathbf{U}_k \mathbf{\Lambda}_k. \tag{4.5}$$

When using the GEVD to find $\mathbf{U}_k$, the projection matrices that are found maximize the trace of Eq. (4.4).

In order to prevent ill-conditioned matrices and make sure the solution converges, a regularization term times the identity matrix has to be added to both scatter matrices:

$$\begin{aligned}
\mathbf{S}_b' &= \mathbf{S}_b + \lambda \mathbf{I}, \\
\mathbf{S}_w' &= \mathbf{S}_w + \lambda \mathbf{I}.
\end{aligned} \tag{4.6}$$

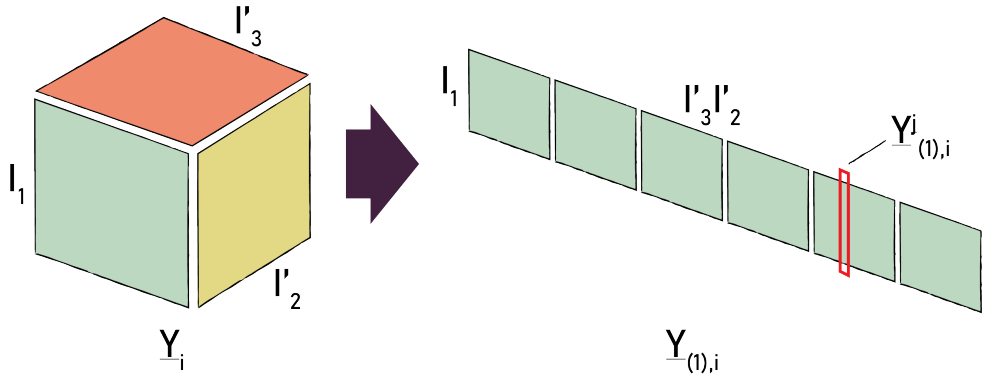

Figure 4.2: Visualization of the vectors used to compute the within- and between-scatter matrices in HODA. This is an example in the case of a third-order tensor, for which $\mathbf{U}_1$ is calculated.

This λ is found by inspecting the values in $\boldsymbol{\lambda}_k$ and the convergence of $\mathbf{U}_1, \ldots, \mathbf{U}_K$. This results in a range of possible values that seem suitable; these values are then used in a parameter sweep to find the λ for which the classification rate is the highest.

Algorithm 1 Higher Order Discriminant Analysis

Input: $\mathbf{c} \in \mathbb{R}^{I_0}$, $\underline{\mathbf{X}} \in \mathbb{R}^{I \times I_1 \times \ldots \times I_K}$, $\mathbf{R}$, λ, stopping criterion
Output: $\underline{\mathbf{X}}$', $\mathbf{U}_1 \in \mathbb{R}^{I_1 \times I'_1}$,...,$\mathbf{U}_K \in \mathbb{R}^{I_K \times I'_K}$
Initialisation : $\mathbf{U}_1 \rightarrow \mathbf{I} \in \mathbb{R}^{I_1 \times I'_1}$,...,$\mathbf{U}_K \rightarrow \mathbf{I} \in \mathbb{R}^{I_K \times I'_K}$
while stopping criterion is not reached **do**
  for $k = 1$ to K **do**
    $\underline{\mathbf{Y}} = \underline{\mathbf{X}} \times_1 \mathbf{U}_1 \times_2, \ldots, \times_{k-1} \mathbf{U}_{k-1} \times_{k+1} \mathbf{U}_{k+1}, \ldots, \times_K \mathbf{U}_K$
    $\mathbf{S}_b = \sum_{j=1}^{\prod_{o \neq k} I_o} \mathbf{S}_b^j, \quad \mathbf{S}_b^j = \sum_{c=1}^{N_c} n_c (\bar{\mathbf{Y}}_{(k),c}^j - \bar{\mathbf{Y}}_{(k)}^j)(\bar{\mathbf{Y}}_{(k),c}^j - \bar{\mathbf{Y}}_{(k)}^j)^T$
    $\mathbf{S}_w = \sum_{j=1}^{\prod_{o \neq k} I_o} \mathbf{S}_w^j, \quad \mathbf{S}_w^j = \sum_{i=1}^{I} (\mathbf{Y}_{(k),i}^j - \bar{\mathbf{Y}}_{(k),c_i}^j)(\mathbf{Y}_{(k),i}^j - \bar{\mathbf{Y}}_{(k),c_i}^j)^T$
    $\mathbf{S}_b = \mathbf{S}_b + \lambda \mathbf{I}$, $\mathbf{S}_w = \mathbf{S}_w + \lambda \mathbf{I}$
    Solve: $\mathbf{S}_b \mathbf{U}_k = \mathbf{S}_w \mathbf{U}_k \boldsymbol{\Lambda}_k, \quad \mathbf{U}_k \in \mathbb{R}^{I_k \times I'_k}$, for $\mathbf{U}_k$
  end for
end while
$\underline{\mathbf{X}}'_i = \underline{\mathbf{X}}_i \times_1 \mathbf{U}_1 \times_2, \ldots, \times_K \mathbf{U}_K$

The ranks $\mathbf{R}$ in this algorithm that determine the size of the matrices $\mathbf{U}_1, \ldots, \mathbf{U}_K$ can be either set as an input or be determined based on the eigenvalues found in $\boldsymbol{\Lambda}_k$. This second method is used in the implementation of this thesis. When inspecting the eigenvalues, they are mostly the same value except for a few lower ones and a few higher ones. The eigenvalue amount higher than the median value determines the rank I'_k, and the corresponding eigenvectors are used to construct $\mathbf{U}_k$.

In order to have Matlab perform the algorithm described by [74] quicker, another implementation is made for this thesis. The scatter matrices can also be computed using matrix products instead of a large number of summations. To do this a matrix $\mathbf{C} \in \mathbb{R}^{I \times C}$ is made:

$$\mathbf{C} = \begin{bmatrix} \frac{1}{n_1}\mathbf{c}_1 & \frac{1}{n_2}\mathbf{c}_2 & \ldots & \frac{1}{n_C}\mathbf{c}_C \end{bmatrix}, \tag{4.7}$$

where $\mathbf{c}_c$ are column vectors of length I, with ones if element i belongs to class c and zeros otherwise. The input tensors $\underline{\mathbf{Y}}_i$ are all concatenated for the faster implementation, creating a tensor $\underline{\mathbf{Y}} \in \mathbb{R}^{I \times I'_1 \times \ldots \times I_k \times \ldots \times I'_K}$. Using the matrix $\mathbf{C}$ and the mode-$1, k$-matricization of $\underline{\mathbf{Y}}$, tensor $\underline{\mathbf{Y}}_{(1,k)}$, it is

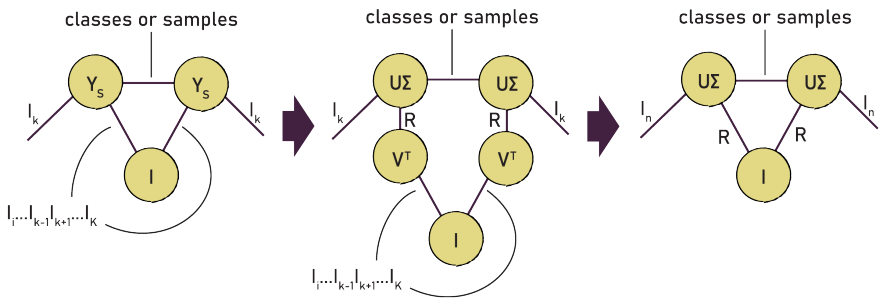

Figure 4.3: Applying singular value decomposition can be used to reduce computational complexity to find scatter matrices

possible to efficiently calculate two tensors $\underline{\mathbf{Y}}_{Sb}$ and $\underline{\mathbf{Y}}_{Sw}$:

$$\begin{aligned}
\underline{\mathbf{Y}}_C &= \underline{\mathbf{Y}}_{(1,k)} \times_1 \mathbf{C} \\
\underline{\mathbf{Y}}_{S_b} &= \underline{\mathbf{Y}}_C - \underline{\mathbf{Y}}_{(1,k)} \times_1 [\frac{1}{N}\mathbf{1} \dots \frac{1}{N}\mathbf{1}], \quad \text{with } [\frac{1}{N}\mathbf{1} \dots \frac{1}{N}\mathbf{1}] \in \mathbb{R}^{N \times C} \\
\underline{\mathbf{Y}}_{S_w} &= \underline{\mathbf{Y}}_{(1,k)} - \underline{\mathbf{Y}}_C \times_1 [\mathbf{c}_1, \mathbf{c}_2, ..., \mathbf{c}_C]^T.
\end{aligned} \tag{4.8}$$

These tensors can be used to compute the scatter matrices as follows:

$$\begin{aligned}
\mathbf{S}_b &= (\underline{\mathbf{Y}}_{S_b} \times_3 \underline{\mathbf{Y}}_{S_b}) \times_2^3 \mathbf{I} \\
\mathbf{S}_w &= (\underline{\mathbf{Y}}_{S_w} \times_3 \underline{\mathbf{Y}}_{S_w}) \times_2^3 \mathbf{I}
\end{aligned} \tag{4.9}$$

Solving these equations can still be computationally complex when the third mode of the tensors $\underline{\mathbf{Y}}_{Sb}$ and $\underline{\mathbf{Y}}_{Sw}$ is large. The length of this mode is $\prod_{o \neq k} I_o$. In these cases, the size can be reduced by performing a singular value decomposition on $\underline{\mathbf{Y}}_{(1,k)}$, this results in the same scatter matrices as illustrated by Fig. 4.3. This method uses the fact that:

$$\mathbf{V}^T \mathbf{I} \mathbf{V} = \mathbf{I}, \tag{4.10}$$

because $\mathbf{V}^T$ is orthonormal. Due to the implementation of certain functions with varying computational complexities, it is hard to determine the computational benefit of this extra step, computing the singular value decomposition (SVD). By comparing the average time it takes to run each version of the algorithm, the singular value decomposition reduced the runtime for applying HODA on the simulations described in the next chapter (Ch. 5). The time difference between the two methods becomes larger when the reduced mode sizes I'_k are larger.

HODA can also be applied to larger tensors that can account for more relations between features and the class. An efficient method to do this is by adding an extra mode to the tensor in which extra features are added. These features can, for instance, be computed using the radial basis function (RBF), which is often used in machine learning to map data to higher non-linear space. This function can be applied to any combination of samples:

$$K(x_i, x_j) = \exp(\frac{(x_i - x_j)^2}{\sigma}), \tag{4.11}$$

where sigma is a parameter that can be tuned. In this thesis, σ will be set to one. If nonlinearities are of importance, it is very likely that this will be the case in the time domain. A tensor that contains this rbf kernel can be constructed by applying the Kernel function to each row of $\underline{\mathbf{X}}_{(t)}$.

Another interesting method of altering the input, is by applying the wavelet transform. This transform is a bit like the short-time Fourier transform in the sense that it can find frequency components at certain times. The advantage of the Wavelet transform over the Fourier transform is that it considers the trade-off between accuracy in the time or frequency domain by finding high frequencies with high time precision and low frequencies with low time time precision. The implementation used for this project is the continuous wavelet transform function in Matlab, “cwt()”, with the default settings. The frequency bands in the WT are often set so that they overlap with the frequency bands often observed in EEG

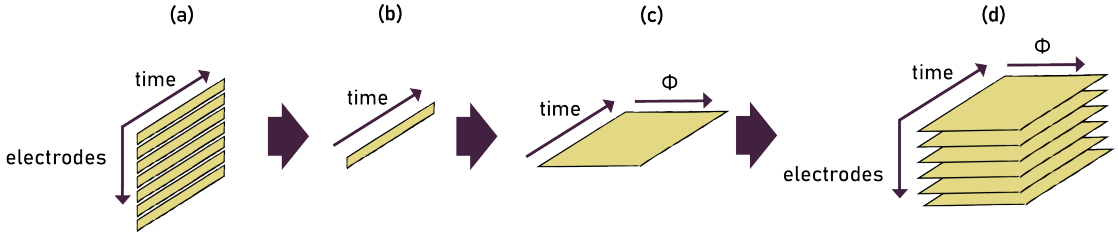

Figure 4.4: Visualisation of how rbf and Wavelet are incorporated in tensor. (a) The tensor for a subject. (b) The vector for a single electrode. (c) The Wavelet or rbf can be applied to the vector, resulting in a matrix. (d) For each electrode the matrices can be concatenated, resulting in a tensor with an extra mode.

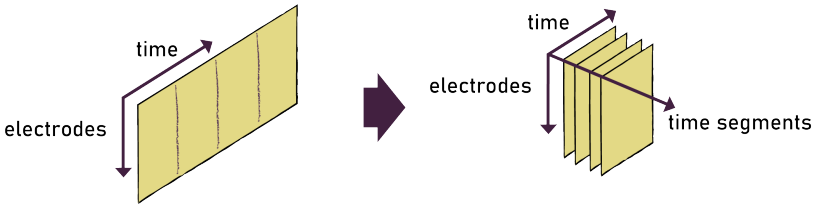

Figure 4.5: Example of how a mode-2 tensor, or matrix, is segmented in the time domain. This results in a mode-3 tensor.

research, but due to the averaging of the trials, these waves are averaged out. These operations can only be applied to vectors. How they are implemented in the case of a mode-2 tensor is shown in Fig. 4.4. This method can also be applied to higher-order tensors; the time vector in the second step is then not only taken for each electrode but also for each electrode-stimulus combination.

A final method used to change the input tensor in this thesis is the use of segmentation in the time domain. Boussé et al. [87] state that this can be useful for large tensors. In this research, it is not discussed whether this is also useful in combination with HODA. An illustration of how this segmentation is done can be seen in Fig. 4.5 for the case of mode-2 tensor. This operation results in an extra mode to the tensor, but the amount of entries in the tensor remains the same.

4.1.2. Sparse HODA

The expected differences in the ERP effect between different disorders are likely to occur in specific components of the ERP waveform. Time samples of brain regions that are unrelated to these differences should theoretically not be taken into account when transforming the tensor to lower dimensions. To make sure these irrelevant features will go to zero, a sparsity constraint can be introduced as formulated by Wen et. al. [88]:

$$||\mathbf{Q}||_{2,1} = \sum_{i=1}^{k} \sqrt{\sum_{j=1}^{m} q_{i,j}^2}, \tag{4.12}$$

where $\mathbf{Q}$ is the projection matrix used in LDA. This constraint uses the l_1-norm, which is often used to induce sparsity [89], on the l_2-norm of the rows of $\mathbf{Q}$. The l_2-norm of the rows corresponds to the weight that is given to feature i, so by constraining these norms to be sparse some features will not be given a weight when the data matrix is reduced in size. For this thesis robust sparse linear discriminant analysis (RSLDA) of Wen et al. [88] is combined with HODA developed by Yan et al. [74]. This combination starts by changing the constraint in Eq. 4.12 into:

$$||\mathbf{U}_k||_{2,1} = \sum_{i=1}^{I_k} \sqrt{\sum_{j=1}^{\prod_{o\neq k}^{K} I_o} u_{k_{i,j}}^2}. \tag{4.13}$$

This constraint can be added when one or more of the projection matrices $\mathbf{U}_k$. The matrices $\mathbf{U}_k$ can be iteratively solved in the same manner as the LDA optimization problem in [88] is solved:

$$\begin{aligned}\min_{\mathbf{P}_k,\mathbf{U}_k,\mathbf{E}_k} & \; Tr\left(\mathbf{U}_k^T\left(\mathbf{S}_w - u\mathbf{S}_b\right)\mathbf{U}_k\right) + \lambda_1\|\mathbf{U}_k\|_{2,1} + \lambda_2\|\mathbf{E}\|_1 \\ s.t. & \; \mathbf{Y}_{(k)} = \mathbf{P}_k\mathbf{U}_k^T\mathbf{Y}_{(k)} + \mathbf{E}_k, \quad \mathbf{P}_k^T\mathbf{P}_k = \mathbf{I},\end{aligned}$$

where $\mathbf{S}_w$ and $\mathbf{S}_b$ are defined by Eq. 4.3, $\mathbf{P}_k$ matrices that can transform the reduced tensor back to the original, and $\mathbf{E}_k$ is an error term. The trace operator and the function to which it is applied in the objective function is a rewritten form of Eq. 4.4, with u added as a parameter to balance the importance of the two scatter matrices. A solution to this problem can be found by transforming it into its Lagrangian form:

$$\begin{aligned}L(\mathbf{P}_k,\mathbf{Q}_k,\mathbf{E}_k,\boldsymbol{\nu}_k) = & \; Tr(\mathbf{U}_k^T(\mathbf{S}_w - u\mathbf{S}_b)\mathbf{U}_k) + \lambda_1||\mathbf{U}_k||_{2,1} + \lambda_2||\mathbf{E}_k||_1 \\ & + \langle\boldsymbol{\nu}_k, \mathbf{Y}_{(k)} - \mathbf{P}_k\mathbf{U}_k^T\mathbf{Y}_{(k)} - \mathbf{E}_k\rangle \\ & + \frac{\beta}{2}||\mathbf{Y}_{(k)} - \mathbf{P}_k\mathbf{U}_k^T\mathbf{Y}_{(k)} - \mathbf{E}_k||_F^2,\end{aligned} \tag{4.14}$$

where $\boldsymbol{\nu}$ is the Lagrangian multiplier and β a penalty term. A solution to this problem can be found by iteratively solving for $\mathbf{U}_k$, $\mathbf{P}_k$, $\mathbf{E}_k$ and $\mathbf{Y}$. This iteration is repeated each time a matrix $\mathbf{U}_t$, the projection matrix corresponding to the time mode, has to be computed in HODA. Because it is expected that the differences between the classes in the other modes are not sparse, projection matrices corresponding to these modes are found the same way as in HODA. The algorithm described above can be found in algorithm box 2. In this algorithm box, a few variables are present that are not mentioned above. The matrix $\mathbf{D}$ is used to find the projection matrix $\mathbf{U}_k$:

$$\mathbf{D} = \begin{bmatrix} \frac{1}{||\mathbf{u}_1||_2} & 0 & \dots & 0 \\ 0 & \frac{1}{||\mathbf{u}_2||_2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \frac{1}{||\mathbf{u}_{I_k}||_2} \end{bmatrix}, \tag{4.15}$$

where $\mathbf{u}_{i_k}$ are the row vectors of $\mathbf{U}_k$. This matrix $\mathbf{D}$ is used to calculate $\mathbf{U}_k$ as follows:

$$\mathbf{U}_k = (2(\mathbf{S}_w - u\mathbf{S}_b) + \lambda_{sparse}\mathbf{D} + \beta)^{-1}. \tag{4.16}$$

To calculate $\mathbf{P}_k$, The matrix $\mathbf{M}$ is used:

$$\begin{aligned}\mathbf{P}_k &= \mathbf{U}\mathbf{V}^T \\ \mathbf{U}, \Sigma, \mathbf{V}^T &= \text{svd}(\mathbf{M}\mathbf{Y}_{(k)}\mathbf{U}_k) \\ \mathbf{M} &= \mathbf{Y}_{(k)} - \mathbf{E}_k + \frac{\boldsymbol{\nu}_k}{\beta}.\end{aligned} \tag{4.17}$$

This algorithm will be referred to as sparse higher-order discriminant analysis (SHODA) in this thesis.

4.1.3. Block sparse HODA

When there are differences in the EEGs of the different disorders, we expect these to occur during certain components. These components are not only sparse in time but also span several consecutive time samples. This is why the assumption is made that a block sparse constraint on the projection matrix related to time can lead to a more optimal solution. The implementation of this block sparse constraint is very similar to that of the sparse constraint, but $||\mathbf{U}_k||_{2,1}$ is computed in a different manner:

$$||\mathbf{U}_k||_{2,1} = \sum_{b=1}^{B_k}\sqrt{\sum_{s=1}^{S_k}\sum_{j=1}^{\prod_{n\neq I_k}^{N} I_k} u_{k_{i,s,j}}^2}. \tag{4.18}$$

Algorithm 2 Sparse Higher Order Discriminant Analysis (SHODA)

Input: $\mathbf{c} \in \mathbb{R}^I$, $\underline{\mathbf{X}} \in \mathbb{R}^{I \times I_1 \times \ldots \times I_K}$, $\mathbf{R}$, λ, stopping criterion
Output: $\underline{\mathbf{X}}$', $\mathbf{U}_1 \in \mathbb{R}^{I_1 \times I_1'}$,...,$\mathbf{U}_K \in \mathbb{R}^{I_K \times I_K'}$
Initialisation : $\mathbf{U}_1 \rightarrow \mathbf{I} \in \mathbb{R}^{I_1 \times I_1'}$,...,$\mathbf{U}_K \rightarrow \mathbf{I} \in \mathbb{R}^{I_K \times I_K'}$
while stopping criterion is not reached **do**
  for $k = 1$ to K **do**
    $\underline{\mathbf{Y}} = \underline{\mathbf{X}} \times_1 \mathbf{U}_1 \times_2, ..., \times_{k-1} \mathbf{U}_{k-1} \times_{k+1} \mathbf{U}_{k+1}, ..., \times_N \mathbf{U}_K$
    $\mathbf{S}_b = \sum_{j=1}^{\prod_{o \neq k} I_o} \mathbf{S}_b^j, \;\; \mathbf{S}_b^j = \sum_{c=1}^{N_c} n_c (\bar{\mathbf{Y}}_{(k),c}^j - \bar{\mathbf{Y}}_{(k)}^j)(\bar{\mathbf{Y}}_{(k),c}^j - \bar{\mathbf{Y}}_{(k)}^j)^T$
    $\mathbf{S}_w = \sum_{j=1}^{\prod_{o \neq k} I_o} \mathbf{S}_w^j, \;\; \mathbf{S}_w^j = \sum_{i=1}^{I} (\mathbf{Y}_{(k),i}^j - \bar{\mathbf{Y}}_{(k),c_i}^j)(\mathbf{Y}_{(k),i}^j - \bar{\mathbf{Y}}_{(k),c_i}^j)^T$
    $\mathbf{S}_b = \mathbf{S}_b + \lambda \mathbf{I}$, $\mathbf{S}_w = \mathbf{S}_w + \lambda \mathbf{I}$
    if $I_k = I_t$ **then**
      Solve: $\mathbf{S}_b \mathbf{P}_k = \mathbf{S}_w \mathbf{P}_k \mathbf{\Lambda}_k, \;\; \mathbf{P}_k \in \mathbb{R}^{I_k \times I_n'}$
      while stopping criterion is not reached **do**
        Initialisation : $\mathbf{U}_k = 0, \mathbf{E}_k = 0, \beta = 0.1, \rho = 1.01, \beta_{\text{max}} = 10^5, u = 10^{-4}$
        Solve $\mathbf{U}_k = (2(\mathbf{S}_w - u\mathbf{S}_b) + \lambda_{sparse}\mathbf{D} + \beta)^{-1}$
        Solve for $\mathbf{P}_k = \mathbf{U}\mathbf{V}^T$ where $\mathbf{U}$ and $\mathbf{V}^T$ are obtained from SVD($\mathbf{M}\mathbf{Y}_{(k)}\mathbf{U}_k$)
        Solve $\mathbf{E}_k = \text{shrink}(\mathbf{E}_0, e)$
        Solve $\boldsymbol{\nu} = \boldsymbol{\nu} + \beta(\mathbf{Y}_{(k)} - \mathbf{P}\mathbf{U}_k^T\mathbf{Y}_{(k)} - \mathbf{E})$, $\beta = \min(\rho\beta, \beta_{\text{max}})$
      end while
    else
      Solve: $\mathbf{S}_b \mathbf{U}_k = \mathbf{S}_w \mathbf{U}_k \mathbf{\Lambda}_k, \;\; \mathbf{U}_k \in \mathbb{R}^{I_k \times I_n'}$, for $\mathbf{U}_k$
    end if
  end for
end while
$\underline{\mathbf{X}}_i' = \underline{\mathbf{X}}_i \times_1 \mathbf{U}_1 \times_2, ..., \times_K \mathbf{U}_K$

Here, the projection matrix has columns split over B_k blocks of a size of S_k. Using this new constraint can be implemented by changing the SHODA algorithm. The matrix $\mathbf{D}$ has to be calculated using blocks of the matrix $\mathbf{U}_k$, that are written as $\mathbf{U}_b$:

$$\mathbf{D} = \begin{bmatrix} \frac{1}{||\mathbf{U}_{k,1}||_F} & 0 & \ldots & 0 \\ 0 & \frac{1}{||\mathbf{U}_{k,1}||_F} & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \frac{1}{||\mathbf{U}_{k,B_k}||_F} \end{bmatrix}. \tag{4.19}$$

The block size is determined by the shortest component length, which is the N1 component in the simulation. This component lasts around 20 ms, or 5 time samples. The resulting algorithm is referred to as block sparse higher order discriminant analysis (BSHODA) in this thesis.

4.2. Tensor Regression

As described in Section 2.3.4, the goal of regression is to find a function that has the best fit for a system's behaviour. This is often done by first selecting a function or a set of functions, and then using training data to optimize the parameters of this function. In the usual case, these functions have a scalar or a vector as input and output, but when data occurs in higher order, this is not ideal. Therefore, tensor regression models have been developed, that exploit the structure of the tensor and tensor decompositions in order to find an appropriate model. In the case of the KHL, the input tensors are the ERP measurements, and the output are scores from other tests. In this chapter, two of these tensor decompositions, the CPD and the TD, for regression are explored. For both these methods, a constraint is also proposed to make the algorithms more suited to the ERP data.

4.2.1. CPD Based Regression

Zhou et al. [81] propose a basic format to make tensor regression models. In their work, a regression model based on a generalized linear model [90] is used with a tensor as input. the model $\hat{y}$ is used to relate the input to an output. This link function has the following format for a linear model using tensors:

$$\hat{y}_i = \alpha + \langle \underline{\mathbf{B}}, \underline{\mathbf{X}}_i \rangle, \tag{4.20}$$

where $\underline{\mathbf{B}}$ contains the model parameters and $\underline{\mathbf{X}}_i$ is the input for a single subject. In [81] the applications in neuroimaging are also taken into account, so an extra term is added to account for variables such as age and gender:

$$\hat{y}_i = \alpha + \boldsymbol{\gamma}^T \mathbf{z}_i + \langle \underline{\mathbf{B}}, \underline{\mathbf{X}}_i \rangle. \tag{4.21}$$

This model now has a lot of weights that need to be optimized; in the case of ERPs, this could, for instance, be 20 *subjects* $\times\, 200$ *time samples* $\times\, 120$ *trials* $= 480,000$ weights. Zhao et al. [81] reduce this by taking a rank R approximation of $\underline{\mathbf{B}}$, resulting in only $R \times (20 + 200 + 120) = 340R$ parameters for the previously mentioned example. To find a good approximation for the factor matrices $\mathbf{B}_1, \mathbf{B}_2, ..., \mathbf{B}_K$ a minimum mean square error (MMSE) estimator is used. All samples are assumed to be independent and identically distributed (i.i.d).

In order to find $\hat{y}$ using the MLE, the factor matrices are optimized individually using a block relaxation algorithm [91] as follows:

$$\hat{y}_i = \alpha + \boldsymbol{\gamma}^T \mathbf{z}_i + \langle \sum_{r=1}^{R} \mathbf{b}_1^{(r)} \circ \mathbf{b}_2^{(r)} \circ ... \circ \mathbf{b}_K^{(r)}, \underline{\mathbf{X}}_i \rangle \tag{4.22}$$

and to optimize for $\mathbf{B}_k$ this can be rewritten to

$$\begin{aligned} \hat{y}_i &= \alpha + \boldsymbol{\gamma}^T \mathbf{z}_i + \langle (\mathbf{B}_K \odot \mathbf{B}_{N-1} \odot ... \odot \mathbf{B}_1)\mathbf{1}_R, \text{vec}(\underline{\mathbf{X}}_i) \rangle \\ &= \alpha + \boldsymbol{\gamma}^T \mathbf{z}_i + \langle \mathbf{B}_k, \mathbf{X}_{(n),i}(\mathbf{B}_K \odot ... \odot \mathbf{B}_{k+1} \odot \mathbf{B}_{k-1} \odot ... \odot \mathbf{B}_1) \rangle. \end{aligned} \tag{4.23}$$

This function can now be used to find all parameters using alternating optimisation. Combining alternating least squares with a regularization term to prevent overfitting results in the following problems that can be solved analytically in series:

$$\begin{aligned} &\min_{\alpha}(y_i - \hat{y}_i(\alpha, \mathbf{z}, \boldsymbol{\gamma}, \mathbf{B}_1, ..., \mathbf{B}_K, \underline{\mathbf{X}})^2 + \lambda_\alpha \alpha^2 \\ &\min_{\boldsymbol{\gamma}}(y_i - \hat{y}_i(\alpha, \mathbf{z}, \boldsymbol{\gamma}, \mathbf{B}_1, ..., \mathbf{B}_K, \underline{\mathbf{X}})^2 + \lambda_{\boldsymbol{\gamma}} ||\boldsymbol{\gamma}||_2^2 \\ &\min_{\mathbf{B}_k}(y_i - \hat{y}_i(\alpha, \mathbf{z}, \boldsymbol{\gamma}, \mathbf{B}_1, ..., \mathbf{B}_K, \underline{\mathbf{X}})^2 + \lambda_{\mathbf{B}} ||\mathbf{B}_k||_2^2. \end{aligned} \tag{4.24}$$

These equations are written for the case of one subject, but to apply regression more subjects are used. This can be done by concatenating outputs $\hat{y}_i$ in vector $\mathbf{y}$ and concatenating the vectors of variables $\mathbf{z}_i$ into a matrix $\mathbf{Z}$. The input tensors for multiple subjects are combined using Eq. 4.25 and Eq. 4.26. The solutions to each of these problems can be found by vectorizing the dot product terms, and for simplicity rewriting the combination of the input and CPD terms as:

$$\mathbf{p}_{k,i} = \text{vec}(\mathbf{X}_{(n),i}(\mathbf{B}_K \odot ... \odot \mathbf{B}_{k+1} \odot \mathbf{B}_{k-1} \odot ... \odot \mathbf{B}_1)), \tag{4.25}$$

for one subject i. These can be stacked for I input subjects as:

$$\mathbf{P}_k = \begin{bmatrix} \mathbf{p}_{k,1} & \mathbf{p}_{k,2} & \cdots & \mathbf{p}_{k,I} \end{bmatrix} \tag{4.26}$$

To obtain the parameters, the derivative has to be found for each objective function, and that derivative is equal to zero. This results in the following equations:

$$\alpha = \frac{1}{I}(\mathbf{y} - \boldsymbol{\gamma}\mathbf{Z} - \langle (\mathbf{B}_K \odot \mathbf{B}_{K-1} \odot ... \odot \mathbf{B}_1)\mathbf{1}_R, \text{vec}(\underline{\mathbf{X}}) \rangle)^T \mathbf{1}/(1 + \lambda_\alpha), \tag{4.27}$$

$$\boldsymbol{\gamma} = (\mathbf{Z}^T\mathbf{Z} + \lambda_z)^{-1}\mathbf{Z}(\mathbf{y} - \alpha\mathbf{1} - \langle (\mathbf{B}_K \odot \mathbf{B}_{K-1} \odot ... \odot \mathbf{B}_1)\mathbf{1}_R, \text{vec}(\underline{\mathbf{X}}) \rangle), \tag{4.28}$$

$$\mathbf{B}_k = (\mathbf{P}_k^T\mathbf{P}_k + \lambda_B)^{-1}\mathbf{P}_k(\mathbf{y} - \alpha\mathbf{1} - \boldsymbol{\gamma}\mathbf{Z}), \tag{4.29}$$

where I is the amount of subjects used to train the regression model, the method to find the regression parameters using CPD is summarized in algorithm 3.

Algorithm 3 CPD based regression

Input: $\mathbf{y}$, $\underline{\mathbf{X}}$, λ_α, $\lambda_{\boldsymbol{\gamma}}$, $\lambda_{\mathbf{B}}$, stopping criterion
Output: α, $\boldsymbol{\gamma}$, $\mathbf{B}_1$,...,$\mathbf{B}_k$
Initialisation : α, $\boldsymbol{\gamma}$, $\mathbf{B}_1$,...,$\mathbf{B}_1$

while stopping criterion is not reached **do**
  for $i = 1$ to I **do**
    $d_i = \langle (\mathbf{B}_K \odot \mathbf{B}_{N-1} \odot ... \odot \mathbf{B}_1)\mathbf{1}_R, \text{vec}(\underline{\boldsymbol{X}}_i) \rangle$
  end for
  $\mathbf{d} = [d_1, \ldots, d_I]$
  $\alpha = \frac{1}{M(1+\lambda_\alpha)}(\mathbf{y} - \boldsymbol{\gamma}\mathbf{Z} - \mathbf{d})^T\mathbf{1}$
  $\boldsymbol{\gamma} = (\mathbf{z}^T\mathbf{z} + \lambda_z)^{-1}\mathbf{z}(\mathbf{y} - \alpha\mathbf{1} - \mathbf{d})$
  for $k = 1$ to K **do**
    for $i = 1$ to I **do**
      $\mathbf{p}_{k_i} = \text{vec}(\mathbf{X}_{i(k)}(\mathbf{B}_K \odot ... \odot \mathbf{B}_{k+1} \odot \mathbf{B}_{k-1} \odot ... \odot \mathbf{B}_1))$
    end for
    $\mathbf{P}_k = [\mathbf{p}_{k_1}, ..., \mathbf{p}_{k_I}]$
    $\mathbf{B}_k = (\mathbf{P}_k^T\mathbf{P}_k + \lambda_B)^{-1}\mathbf{P}_k(\mathbf{y} - \alpha\mathbf{1} - \boldsymbol{\gamma}\mathbf{Z})$
  end for
end while

4.2.2. Time Sparse CPD Based Regression

Many variables, such as age and cognitive functions, result in certain peaks of the ERP being reduced or delayed, according to the literature described in the first chapter. This means that the input tensor only contains features relevant to the output at limited time instances. The weights for features outside these relevant time samples should be approximately zero. This can be enforced by adding a sparsity constraint. This constraint is an addition to the method described by [81] made for this thesis. Because the sparsity is most expected in the time domain, this constraint can be added when finding the CPD factor matrix representing time:

$$\begin{aligned} \min_{\mathbf{B}_t} \ & (\hat{y} - \alpha - \boldsymbol{\gamma}^T\mathbf{z} - \langle \mathbf{B}_t, \mathbf{X}_{i(t)}(\mathbf{B}_K \odot ... \odot \mathbf{B}_{t+1} \odot \mathbf{B}_{t-1} \odot ... \odot \mathbf{B}_1) \rangle + \lambda_B ||\mathbf{B}_t||_2^2 \\ & \text{s.t.} \ \ ||\mathbf{B}_t||_0 < R \end{aligned} \tag{4.30}$$

Where R is the maximum amount of time samples relevant for regression. This problem has no easy solution, but can be relaxed to a convex problem using the l_1-norm:

$$\begin{aligned} \min_{\mathbf{B}_t} \ & (\hat{y} - \alpha - \boldsymbol{\gamma}^T\mathbf{z} - \langle \mathbf{B}_t, \mathbf{X}_{i(t)}(\mathbf{B}_K \odot ... \odot \mathbf{B}_{t+1} \odot \mathbf{B}_{t-1} \odot ... \odot \mathbf{B}_1) \rangle \\ & + \lambda_B ||\mathbf{B}_t||_2^2 + \lambda_{\text{t-sparse}} ||\mathbf{B}_t||_1. \end{aligned} \tag{4.31}$$

This convex problem has no analytical solution but can be solved using CVX, a package for solving convex programs [92, 93]. The resulting algorithm is referred to as sparse canonical polyadic decomposition (SCPD) regression in this thesis.

4.2.3. TD Based Regression

An alternative approach is given by Xiaoshan et al. [83], in which the low-rank approximation is not in the form of a CPD, but of a TD. This results in a model:

$$\hat{y}_i = \alpha + \boldsymbol{\gamma}^T\mathbf{z} + \langle \sum_{r_1=1}^{R_1} ... \sum_{r_K=1}^{R_K} g_{r_1,...,r_N} \mathbf{b}_1^{(r_1)} \circ ... \circ \mathbf{b}_K^{(r_K)}, \underline{\mathbf{X}}_i \rangle, \tag{4.32}$$

where $\mathbf{b}_n^{(r_n)}$ are column vectors of the factor matrices and $g_{r_1,...,r_N}$ are the entries of core tensor $\underline{\mathbf{G}}$. A solution to this problem can also be found iteratively for each $\mathbf{B}_k$, by rewriting the dot product as:

$$\langle \underline{\mathbf{B}}, \underline{\mathbf{X}} \rangle = \langle \mathbf{B}_{(n)}, \mathbf{X}_{(n)}(\mathbf{B}_K \otimes ... \otimes \mathbf{B}_{k+1} \otimes \mathbf{B}_{k-1} \otimes ... \otimes \mathbf{B}_1)\mathbf{G}_{(n)}^T \rangle. \tag{4.33}$$

Using TD increases computational costs, but Xiaoshan et al. argue that their method can still be beneficial due to the increased flexibility of Tucker over CPD [83]. This algorithm is described in algorithm box 4.

Algorithm 4 TD based regression

Input: $\mathbf{y}$, $\underline{\mathbf{X}}$, λ_α, $\lambda_{\boldsymbol{\gamma}}$, λ_G, $\lambda_{\mathbf{B}}$, stopping criterion
Output: α, $\boldsymbol{\gamma}$, $\mathbf{B}_1$,...,$\mathbf{B}_k$
Initialisation : α, $\boldsymbol{\gamma}$, $\mathbf{B}_1$,...,$\mathbf{B}_1$
while stopping criterion is not reached **do**
  for $i = 1$ to I **do**
    $d_i = \langle \underline{\boldsymbol{G}} \times_1 \mathbf{B}_1 \times_2 \mathbf{B}_2 \times_3 ... \times_N \mathbf{B}_K, \underline{\mathbf{X}}_i \rangle$
  end for
  $\mathbf{d} = [d_1, \ldots, d_I]$
  $\alpha = \frac{1}{M(1+\lambda_\alpha)}(\mathbf{y} - \boldsymbol{\gamma}\mathbf{Z} - \mathbf{d})^T\mathbf{1}$
  $\boldsymbol{\gamma} = (\mathbf{z}^T\mathbf{z} + \lambda_z)^{-1}\mathbf{z}(\mathbf{y} - \alpha\mathbf{1} - \mathbf{d})$
  for $k = 1$ to K **do**
    for $i = 1$ to I **do**
      $\mathbf{p}_{k_i} = \text{vec}(\mathbf{X}_{i(k)}(\mathbf{B}_K \otimes ... \otimes \mathbf{B}_{k+1} \otimes \mathbf{B}_{k-1} \otimes ... \otimes \mathbf{B}_1)\mathbf{G}_{(n)})$
    end for
    $\mathbf{P}_k = [\mathbf{p}_{k_1}, ..., \mathbf{p}_{k_I}]$
    $\mathbf{B}_k = (\mathbf{P}_k^T\mathbf{P}_k + \lambda_B)^{-1}\mathbf{P}_k(\mathbf{y} - \alpha\mathbf{1} - \boldsymbol{\gamma}\mathbf{Z})$
  end for
  $\mathbf{P}_G = \mathbf{X}_{(1)}(\mathbf{B}_K \otimes ... \otimes \mathbf{B}_{k+1} \otimes \mathbf{B}_{k-1} \otimes ... \otimes \mathbf{B}_1)$
  $\text{vec}(\underline{\mathbf{G}}) = (\mathbf{P}_G^T\mathbf{P}_G + \lambda_G)^{-1}\mathbf{P}_G(\mathbf{y} - \alpha\mathbf{1} - \boldsymbol{\gamma}\mathbf{Z})$
end while

4.2.4. Time Sparse TD Based Regression

Just like in the CPD regression, it can be beneficial to add a sparsity constraint to one or more of the factor matrices when using the TD for regression. That is why, for this thesis, an alternative method to that of Xiaoshan et al. [83] is proposed that solves with the inclusion of a sparsity constraint. When this regards the factor matrix corresponding to the time mode, the problem of finding this matrix is the following:

$$\begin{aligned} \min_{\mathbf{B}_t} \ & (\hat{y} - \alpha - \boldsymbol{\gamma}^T\mathbf{z} - \langle \mathbf{B}_t, \mathbf{X}_{(t)}(\mathbf{B}_K \otimes ... \otimes \mathbf{B}_{t+1} \otimes \mathbf{B}_{t-1} \otimes ... \otimes \mathbf{B}_1)\mathbf{G}_{(t)}^T \rangle + \lambda_B ||\mathbf{B}_t||_2^2 \\ & \text{s.t. } ||\mathbf{B}_t||_0 < R \end{aligned} \tag{4.34}$$

Where R is the maximum amount of time samples relevant for regression. Again, just like in the CPD case, this problem can be relaxed to a convex problem using the l_1-norm:

$$\begin{aligned} \min_{\mathbf{B}_t} \ & (\hat{y} - \alpha - \boldsymbol{\gamma}^T\mathbf{z} - \langle \mathbf{B}_t, \mathbf{X}_{(t)}(\mathbf{B}_K \otimes ... \otimes \mathbf{B}_{t+1} \otimes \mathbf{B}_{t-1} \otimes ... \otimes \mathbf{B}_1)\mathbf{G}_{(t)}^T \rangle)^2 \\ & + \lambda_B ||\mathbf{B}_t||_2^2 + \lambda_{\text{t-sparse}} ||\mathbf{B}_t||_1. \end{aligned} \tag{4.35}$$

The TD regression algorithm can easily be altered to solve this convex problem instead of finding the analytical unconstrained solution. This is done using CVX [92, 93]. The resulting algorithm is referred to as sparse tucker decomposition (STD) regression in this thesis.

5

Simulation Results

Before applying the methods described in Chapter 4 on the real data from the Child Brain Facility, it is useful to test the validity on some simulated data. By using simulated data, all parameters, such as noise levels and ERP amplitude and delays, are controlled. This allows for careful examination of the algorithm's outcomes, especially when investigating the features that are used in the algorithms.

This chapter first describes how simulated data was made using specific software for ERP simulations. These simulations are then used to test both the discriminant analysis and regression methods, and the results are shown in corresponding sections.

5.1. Simulated Data

In order to simulate ERP data in a realistic manner, software designed for this specific goal is used. This software is called BESA Simulator and allows the user to create different dipoles in the brain that create an ERP waveform. The simulator calculates how this waveform is received by each of 32 different electrodes across the scalp and, in addition, adds noise similar to that of real EEG measurements. Of these 32 electrodes, the simulations are used only for those also used in the KHL (Fz, C3, C4, F3, F4). An example of how the dipoles, ERP waveforms, and resulting measurements look can be seen in Fig. 5.1. Matlab was used to automatically generate models with variations in dipole locations, dipole directions, ERP amplitudes, and delays to get realistic simulations. A batch script that was interpretable by the BESA simulator software was used to automically turn these models to simulation data. There are four different types of simulations made to test the algorithms. Simulations were made for both discriminant analysis and regression for both the MMN and ACC. The parameters for each of these types of simulations are described in the following subsections.

5.1.1. Simulation of MMN for Discriminant Analysis

Discriminant analysis is used to differentiate between different classes to which all subjects belong. There are therefore two types of simulations made, each subject has an ERP for both a frequency and duration deviant. The difference between the two classes is that one class shows no "effect" from this deviating frequency and duration, and the other one does. The dipole's locations and directions are the same for each of these classes and are based on the locations of N1 dipoles in the MMN model described by Jemel et al. [94]. The amplitudes and durations are also loosely based on this same literature but are altered to be more simple and similar to the ERP waveforms from the KHL. This results in a simulation of three peaks with their properties for each stimulus in Table 5.1. In addition to what is listed in this table, it is also worth noting that the waveform at the dipole at the right hemisphere is amplified 1.5 times and that the simulation for each subject slightly varies in amplitude and dipole location. The resulting ERP waveforms with additional noise can be found in Fig. 5.2.

The simulations for the different types of stimulation are used to make two different classes. The first class has two regular waveforms, so no MMN effect, while the second class has a frequency deviant and a duration deviant. This is illustrated in Fig. 5.3.

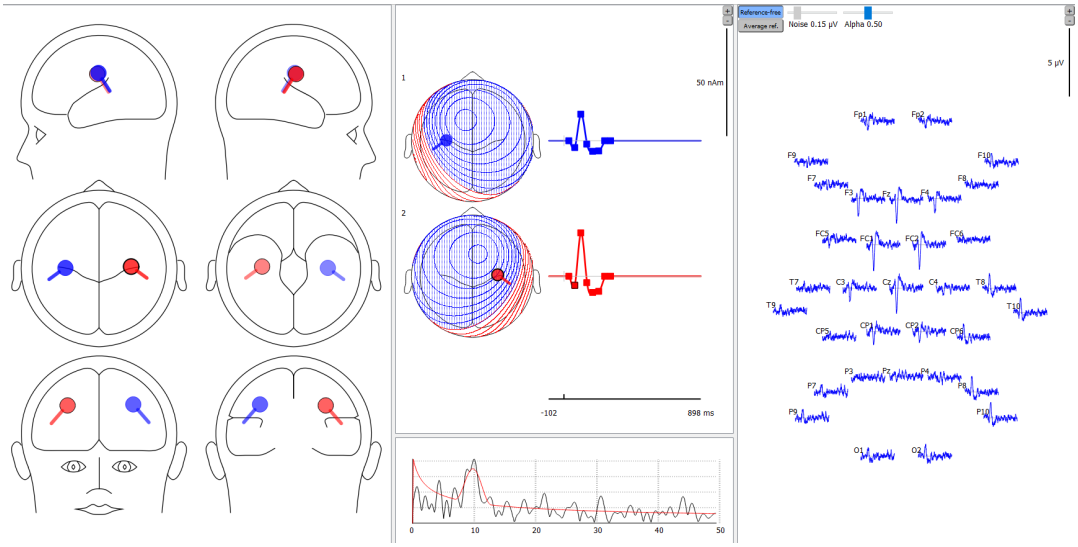

Figure 5.1: Example of auditory event-related potential made in BESA Simulator software used to make MMN simulations. Left: dipole locations and orientation. Middle: waveform generated at each dipole. Right: measurements of the dipoles at 32 electrodes, with noise that has a root mean square (RMS) power of 0.15 the signal amplitude.

Table 5.1: properties of ERP components in the simulation of a regular stimulus, frequency deviant and durations deviant. The component properties where the deviants differ from the regular stimulus are denoted in bold.

Peak	regular stim.	frequency deviant	duration deviant
P1 amplitude	0.2	0.2	0.2
P1 duration	25-85 ms	25-85 ms	**25-90 ms**
N1 amplitude	-1	**0.9**	-1
N1 duration	85-145 ms	85-145 ms	**90-156 ms**
P3 amplitude	0.45	**0.5**	0.45
P3 duration	145-265 ms	145-265 ms	**156-286 ms**

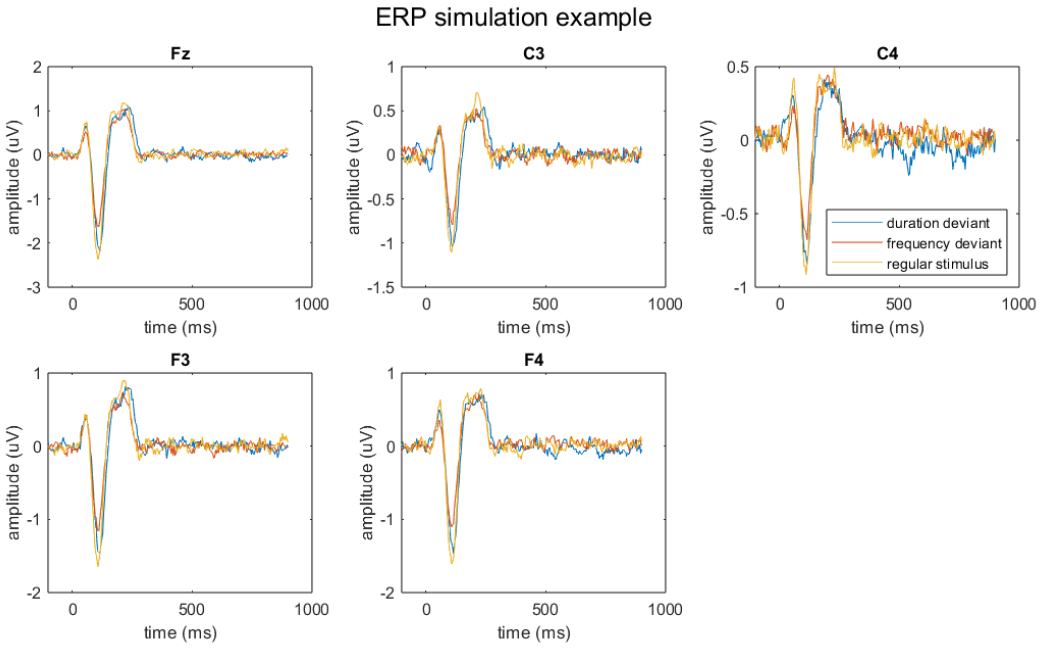

Figure 5.2: MMN waveform with an SNR of 7 dB

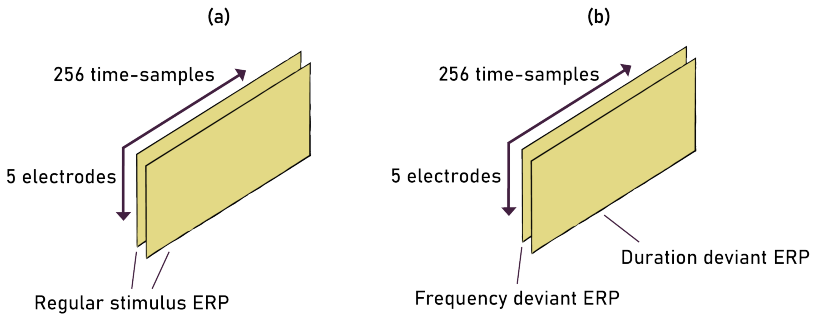

Figure 5.3: Visualization of a tensor $\underline{\mathbf{X}}_i \in \mathbb{R}^{2\times5\times256}$ from a MMN simulation. (a) Tensor for a class where no MMN effect is present. (b) Tensor for a class in which an MMN effect is present.

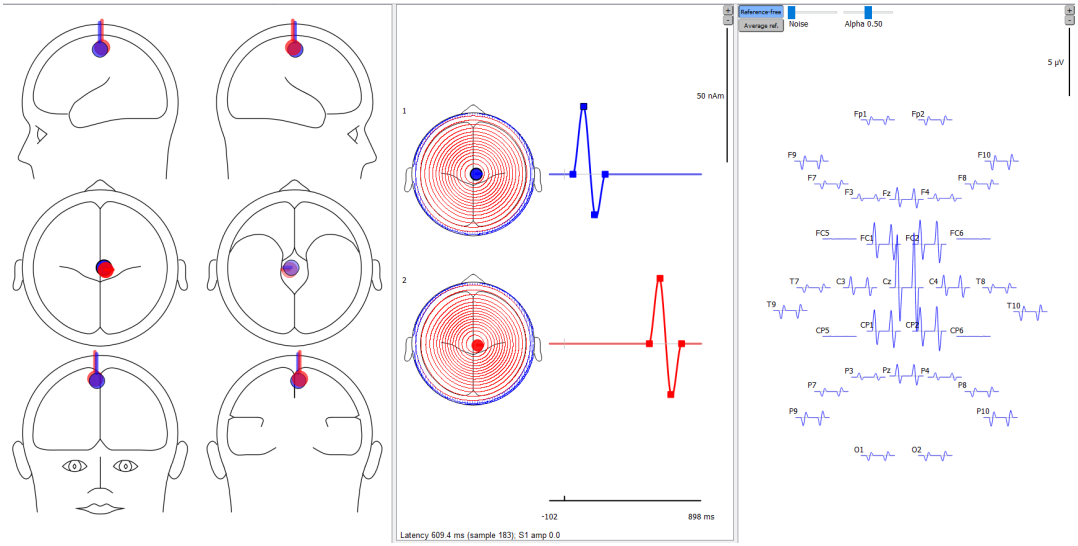

Figure 5.4: Example of ACC model in BESA Simulator. Left: dipole locations and orientation. Middle: waveform generated at each dipole. Right: measurements of the dipoles at 32 electrodes.

5.1.2. Simulation of ACC for Discriminant Analysis

As mentioned before, there is far less research on the ACC than there is on the MMN. This is also the case regarding the exact dipole locations from which the acoustic change complex originates. Because of this, the ACC is modelled a bit simpler, consisting of only an ERP peak originating from the centre of the brain. The location of the dipole varies a bit between subjects and ERP peaks and an example of the model in the BESA Simulator can be seen in Fig. 5.4.

Two different classes are made, with the difference between them in the second peak. In one of the two classes, both peaks have the same mean amplitude. The other class has an amplification of 0.4 times the original amplitude of the second peak. The resulting waveforms for the two different classes are shown in Fig. 5.5.

5.1.3. Simulations for Regression

In order to test the regression methods, the same type of simulation is used as for the classification. Only in order to test regression instead of classification, the amplitudes and delays are not changed according to which class a subject belongs, but these are changed proportionally to some output y, which in the real data can be the severity of the disorders or ability to process sound. The outputs for the subjects are distributed according to a normal distribution:

$$\mathbf{y} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \tag{5.1}$$

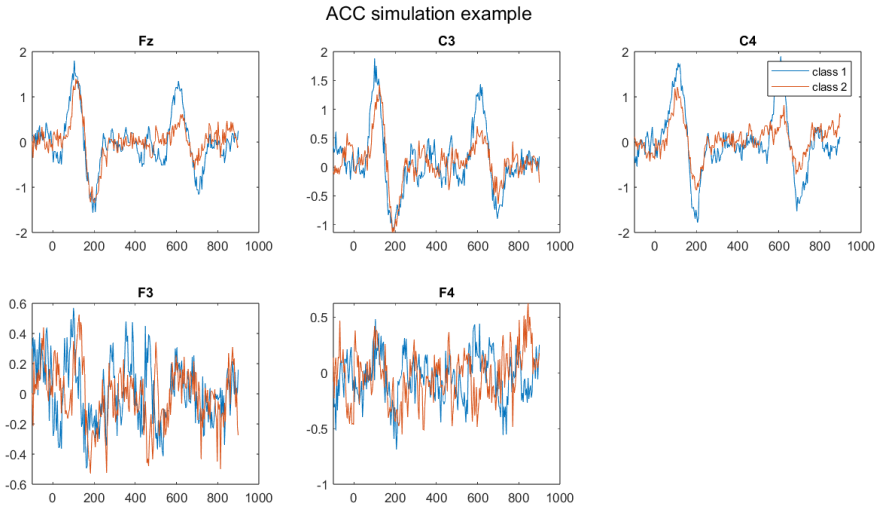

Figure 5.5: ACC waveform for two different classes with an SNR of 7 dB. The first peak, between 50-250 ms, is the same for both classes on average, while the second peak, between 550 and 750 ms, is 0.4 times lower for the second class.

these outputs are then translated to signal amplitudes in both the ACC and MMN experiments by multiplying certain peaks in the simulation by:

$$\mathbf{a} = \mathbf{1} + 0.3 * \mathbf{y}. \tag{5.2}$$

The peaks that are amplified according to these values are the N1 and P3 peak in the MMN simulation (Fig. 5.1) with the regular parameters in Table 5.1, and the second component in the ACC simulation (Fig. 5.4). The goal of the regression algorihms is now to estimate the corresponding y_i for each subject.

5.2. Results Discriminant Analysis

This section will show some results to indicate the performance of the proposed discriminant analysis methods. In order to quantify how well the reduced tensor is after the projection matrices have been applied, classification is applied to these tensors. Besides looking at classification rates, the projection matrices are also analysed to see if it is possible to extract features from these methods.

5.2.1. Classification

Looking at the reduced tensor $\underline{\mathbf{X}}'$ on its own does not tell much about the quality of the discriminant analysis. It is, therefore, useful to apply a simple machine learning algorithm to this reduced tensor and the original tensor to see if performance remains the same or even improves. To obtain reliable classification rates, several rounds of k-fold cross-validation are used in combination with 1-nearest-neighbour classification.

In general, to test a machine learning method's performance, the data is often split up into train- and test data. This way, the weights found by the machine learning algorithm are unbiased regarding the data that it is tested with. To get an accurate classification rate in this manner would require a relatively large amount of test subjects. A method to get more test subjects from the same data size is k-fold cross-validation. In this method, the subjects are split into k groups. This algorithm uses k rounds of applying machine learning, each round with one of the k groups as the test data and the other groups as the train data. The amount of folds used is five because, as a general rule of thumb, five and ten are considered appropriate values. Five is chosen over ten, because of the small size of the dataset, and the amount of test samples is twice as high for 5-fold cross validation.

In the case of the discriminant analysis used in this thesis, the train subjects are used to find projection matrices in each k-fold round. As stated before, these projection matrices don't give any metric

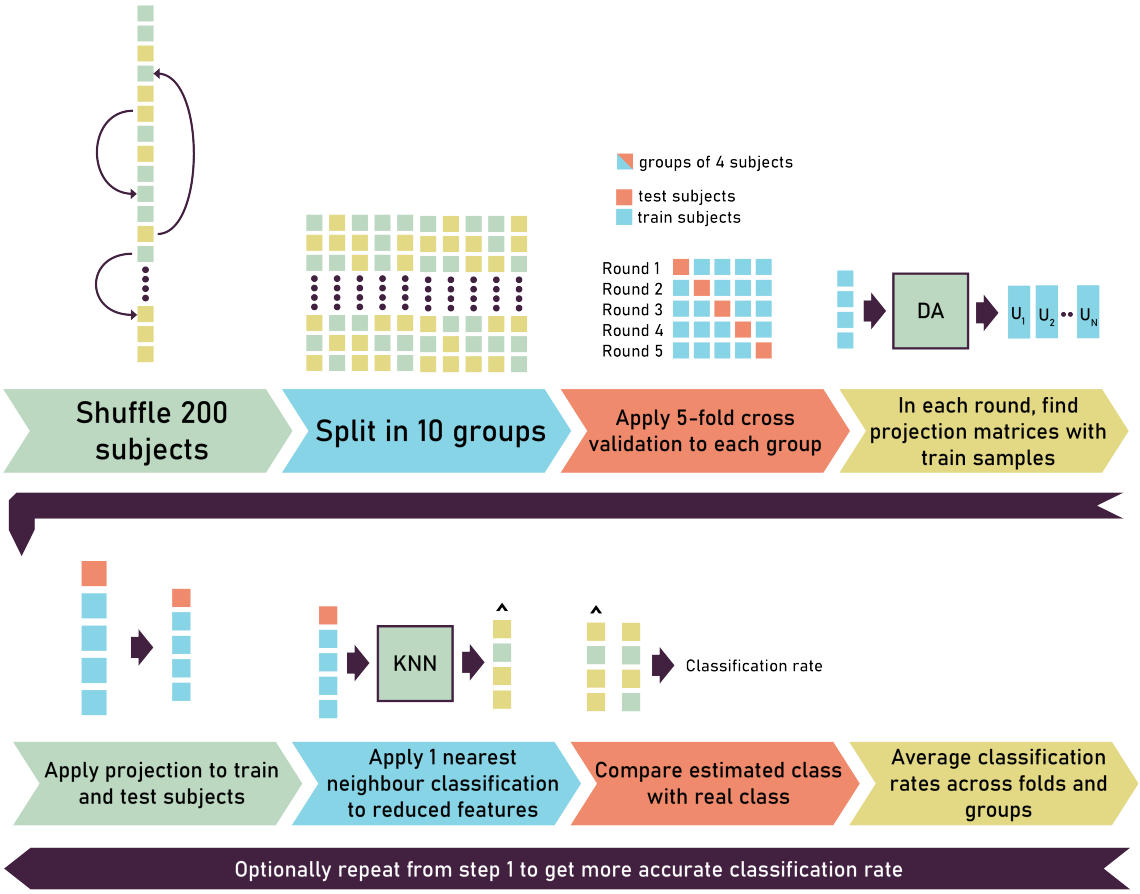

Figure 5.6: Obtaining classification rates visualized in eight steps

to evaluate the validity of the discriminant analysis. To get a sense of performance, the k-nearest neighbour is applied to the reduced train and test subjects. The goal of this machine learning algorithm is to give a good comparison between the different discriminant analysis methods, and not to get the lowest classification rate possible. This is why a simple machine-learning algorithm is sufficient. K-nearest-neighbour is chosen, because it is very easily extended to tensors instead of matrices. As stated before, this ML method uses the distance between the train and test data to find a corresponding class, and the Euclidean distance can also be easily calculated for tensors. 1-nearest neighbour is chosen, because it requires the least amount of computations.

To get a more reliable classification rate, more subjects result in a lower standard error. This is why 100 subjects per class are simulated. This number far exceeds the expected number of subjects from the KHL, so this group of 200 subjects is split up into ten groups of 20 subjects that are used in k-fold cross-validation. Each fold in this cross-validation results in a classification rate. To find a reliable classification rate, the rate of each fold in each group is averaged together. An overview of how the classification rates are obtained can be seen in Fig. 5.6.

Classification rates are obtained not only using the reduced features but also using the full tensor.

5.2.2. Results

The classification rates obtained using the method described above can be found in Fig. 5.7 and 5.8 for the MMN and ACC simulations respectively. These figures show the classification rate for each algorithm, for four different noise levels that are generated using the BESA Simulator. The optimal hyperparameters for each algorithm are found by minimizing the classification error in a parameter

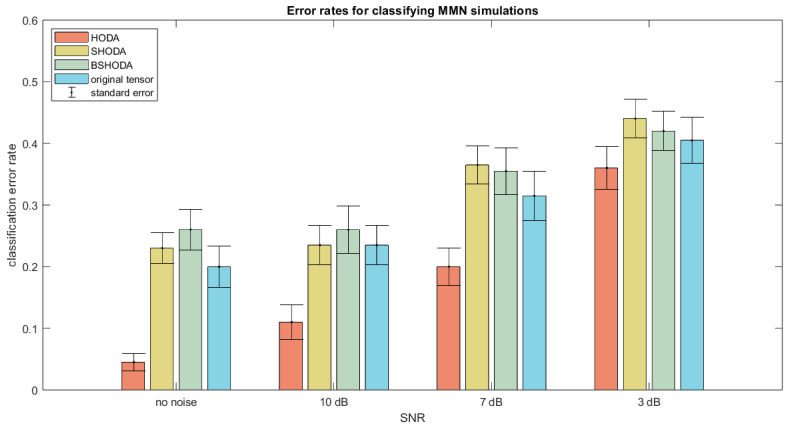

Figure 5.7: Classification error of MMN simulations using different noise levels and different discriminant analysis methods.

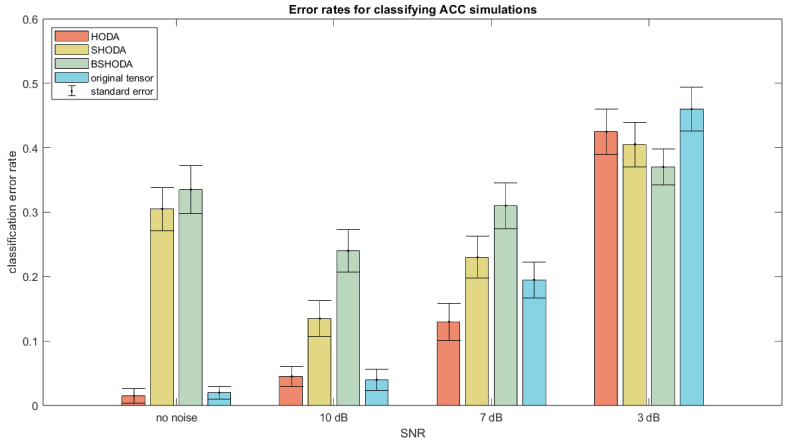

Figure 5.8: Classification error of ACC simulations using different noise levels and different discriminant analysis methods.

sweep for the simulation with an SNR of 7 dB. These same hyperparameters are then used to obtain classification rates for all other noise levels as well.

For the mismatch negativity simulation, the HODA algorithm clearly improves classification performance over the full feature tensors using the 1-nearest neighbour classification. This indicates that the lower dimensional space on which the original tensor is projected is better to discriminate between the different classes. This is, however, not the case for both algorithms that assume sparsity in the projection matrices. The classification errors are the same or slightly higher than for the original tensor, depending on the signal-to-noise ratio.

In the ACC simulation, the difference in performance between HODA and the original tensor is smaller. When looking at the projection matrices, it seems likely that this is due to the algorithm also giving a lot of weight to the first peak in which the signal does not differ. The two algorithms based on sparsity result in a much worse classification rate in the case of no noise. This is likely due to ill-conditioned matrices when little noise is present. For the lowest SNR of 3 dB, SHODA and BSHODA slightly outperform HODA.

To obtain the scatter matrices for these extended tensors, the SVD is used as described in Fig. 4.3. The resulting classification rates of applying HODA to these extended and altered tensors for the MMN simulations are shown in Fig. 5.9. In the 10 and 7 dB, the rbf and wavelet transform appear to improve results, but not by much. The segmentation does, on the other hand, clearly improve the results. In Fig. 5.10 can be seen that the segmentation works even better in the case of the ACC simulation.

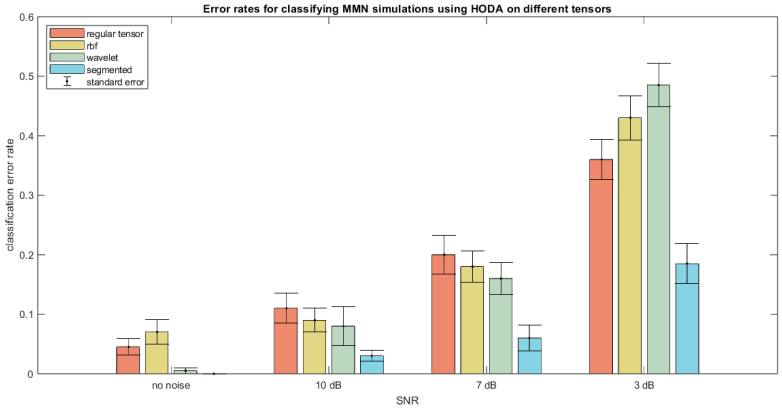

Figure 5.9: Classification rates for HODA and KNN applied to the regular input tensor from the MMN simulation and inputs that have been altered by applying rbf, wt or segmentation.

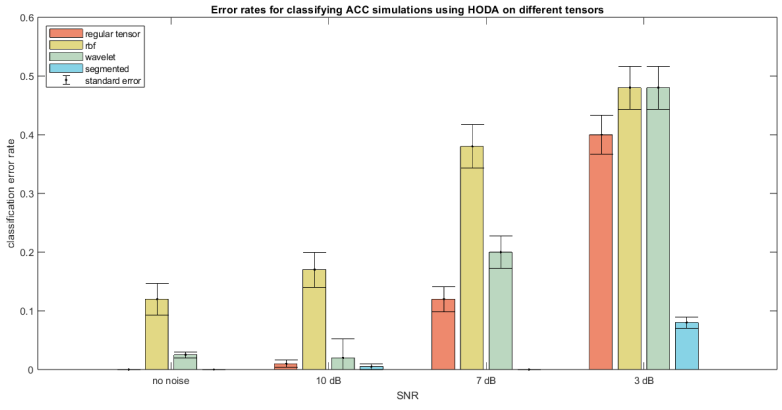

Figure 5.10: Classification rates for HODA and KNN applied to the regular input tensor from the ACC simulation and inputs that have been altered by applying rbf, wt or segmentation.

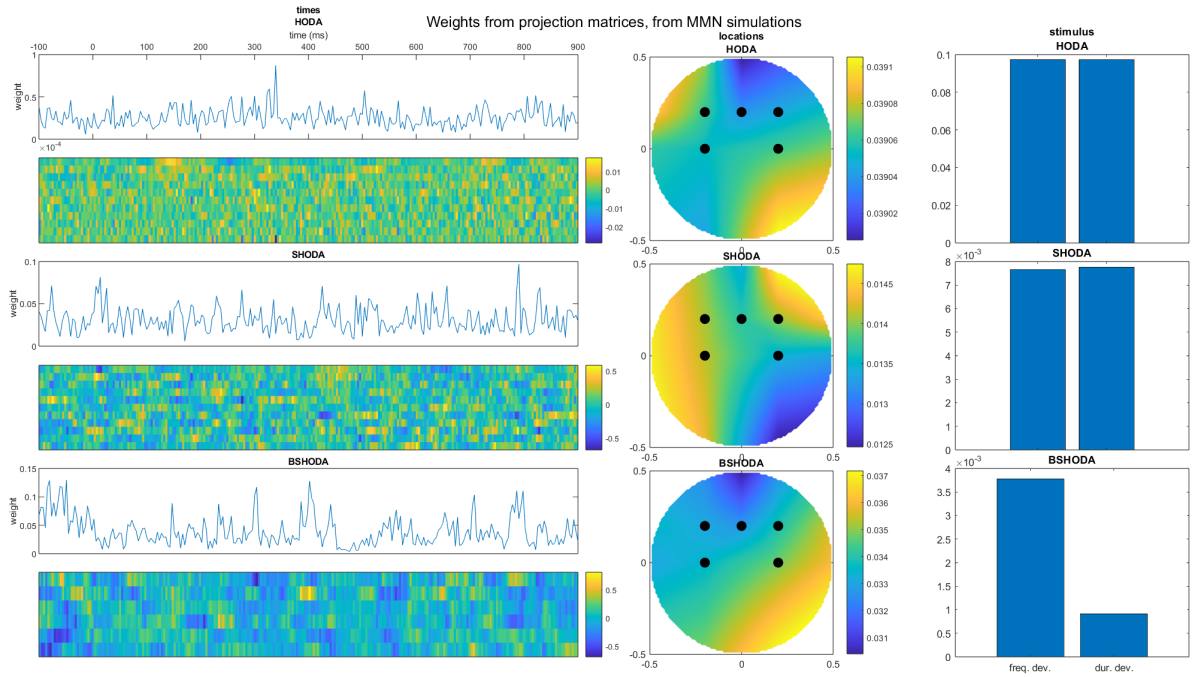

Figure 5.11: Weights extracted from the projection matrices produced by the different DA algorithms on the MMN simulation with a SNR of 7 dB. On the left the weights can be seen that correspond to the projection matrix related to time, with a heatmap of this matrix below. In the middle, the weights corresponding to the locations are plotted on a scalp map. The right shows the weights for the two different stimuli that are used.

5.2.3. Feature Extraction

Checking the classification rates gives a good indication of the ability of discriminant analysis (DA) to find a space in which the classes are better separated. However, what is probably even more relevant for the KHL is to find the underlying mechanisms behind the disorders. We can extract at what times and locations the classes differentiate from each other by looking at the projection matrices. Each row of these matrices corresponds to a certain location of the sample in time, so by summing the squared values of the matrix over the columns, it becomes visible how apparent each sample is in the reduced feature space.

For the feature extraction, only one group of 20 subjects was used, which should be more comparable to the KHL data. When averaging all different waveforms, regular, frequency deviant, and duration deviant, it is possible to look for differences between these waveforms in this group. These waveforms can be seen in Fig. 5.2, for the case where noise is added to get an SNR of 7 dB.

In order to help find when and where the waveforms differentiate it is possible to use the projection matrices by summing over the columns of $\mathbf{U}_k$ which is the same as:

$$\mathbf{w}_k = \text{diag}(\mathbf{U}_k^T \mathbf{U}_k). \tag{5.3}$$

Fig. 5.11 shows this weight vector for all the projection matrices that are found after applying the different DA methods to the MMN simulations, for the same data that was used to obtain Fig. 5.2. This group was used in 5-fold classification, and for each fold the weights were calculated using Eq. 5.3. The weights displayed in Fig. 5.11 are calculated for the projection matrices corresponding to time, location and stimulus. Despite the decent improvement in the classification rate of HODA, no specific times, locations or stimuli are clearly indicated to be of importance by this figure.

The same analysis done on the MMN simulation can also be applied used on the ACC simulation. In Fig. 5.12, the weights extracted from the projection matrices are shown. In this plot, it can be seen that the projection matrix found by HODA corresponding to time has a higher amplitude around 620 ms. This is where the two classes differentiate, so this is what the algorithm should do.

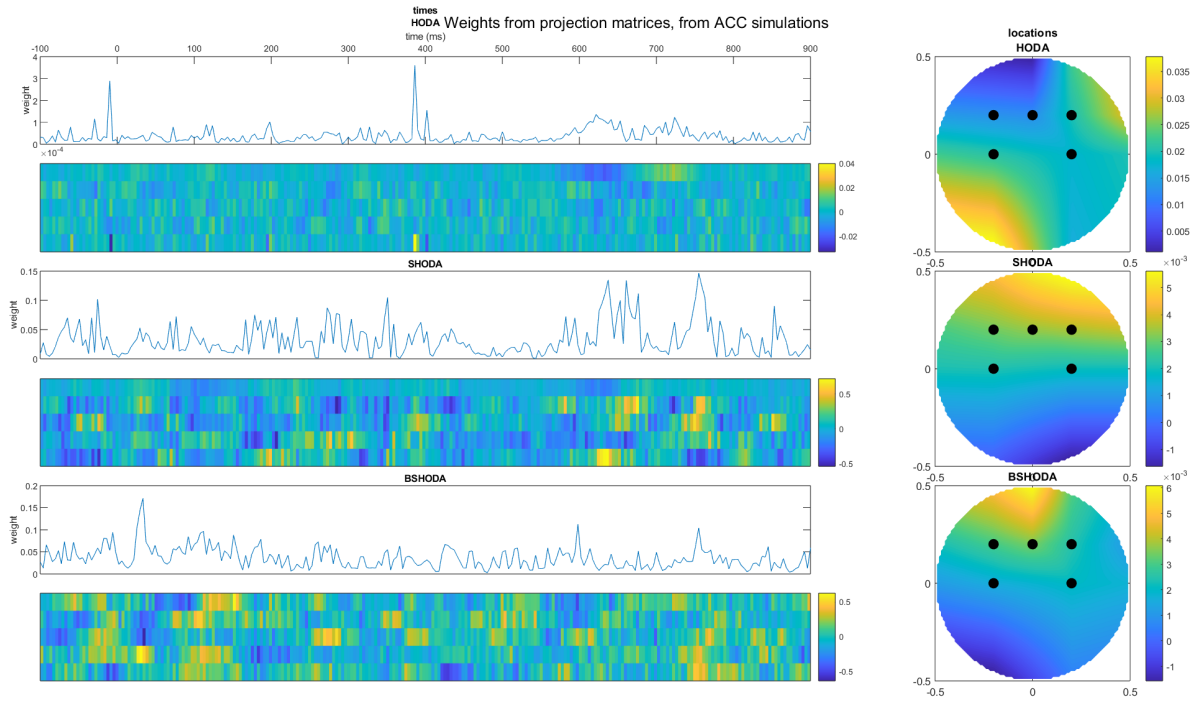

Figure 5.12: Weights extracted from the projection matrices produced by the different DA algorithms on the ACC simulation with a SNR of 7 dB. On the left the weights can be seen that correspond to the projection matrix related to time, with a heatmap of this matrix below. In the middle, the weights corresponding to the locations are plotted on a scalp map. The right shows the weights for the two different stimuli that are used.

When using the wavelet transform to create an extra mode for the input tensor, this results in a projection matrix corresponding to the frequencies in the signal. The weights in this projection matrix are shown in Fig. 5.13. The weights of the frequencies in this plot look fairly similar to a Fourier transform of an EEG or ERP signal.

Weight can also be computed for the projection matrices that are found using the segmented tensor. Especially the mode corresponding to the segments is interesting, as shown in Fig. 5.14 for the MMN simulation. Although the resolution in the time domain is reduced, very clear peaks can now be seen where the two classes differ. In Fig. 5.15, these weights can also be seen for the ACC simulation. Here, it can be clearly seen that the classes differ around the second simulated ACC peak.

5.3. Results Tensor Regression

To validate the usefulness of the proposed regression methods, they are applied to simulated data. The performance of these algorithms is compared using the mean square error. The resulting models based on tensor decompositions are also analysed to find which time samples and locations are being weighted the most to estimate the output.

The mean square error is the performance metric used to evaluate the regression methods. The MSEs are obtained in a similar fashion to how the classification rates are obtained for discriminant analysis by first splitting all the simulations into groups and then applying k-fold cross-validation. Two groups of 50 subjects are made, which are then tested using 5-fold cross-validation. For each estimated test subject, the square of the difference between the estimate and the true output is calculated, and all these values of all groups and folds are averaged to obtain the mean square error.

5.3.1. Results

In Fig. 5.16 and 5.17, the resulting mean square errors for each method for different SNRs are shown. In the case of low noise and the MMN simulation, the regression model is able to predict the output with a relatively low error. An example of how these estimates compare to the real output can be seen

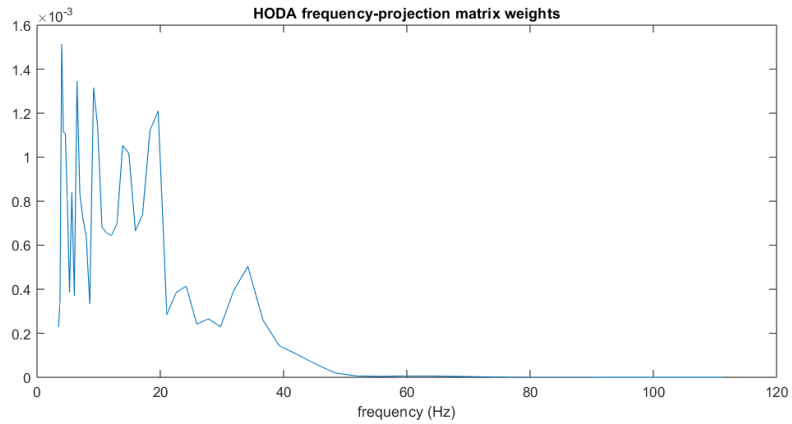

Figure 5.13: Weights corresponding to different frequencies in the projection matrix that are created by applying HODA on tensor with wavelet transform.

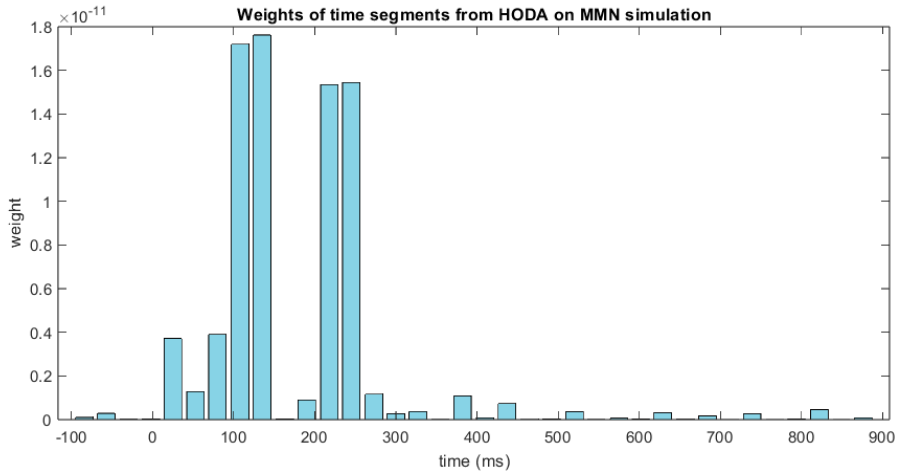

Figure 5.14: Weights found from the projection matrix corresponding to time segments.

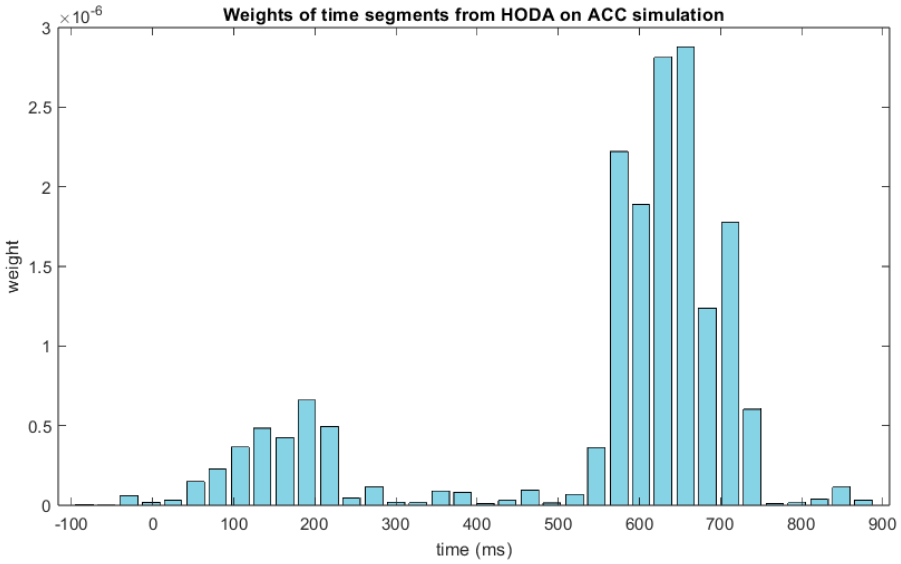

Figure 5.15: Weights found from the projection matrix corresponding to time segments.

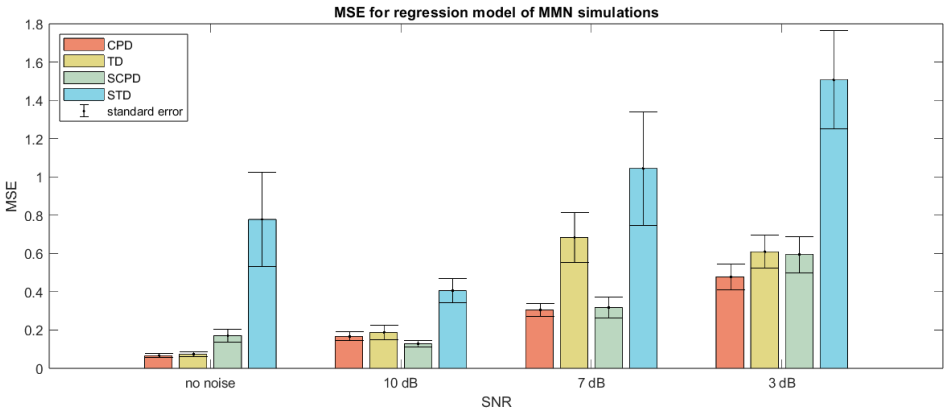

Figure 5.16: Results from applying the regression algorithms to the MMN simulations. The mean square error is calculated using predicted outputs from the regression models and the real simulated outputs using 5-fold cross-validation.

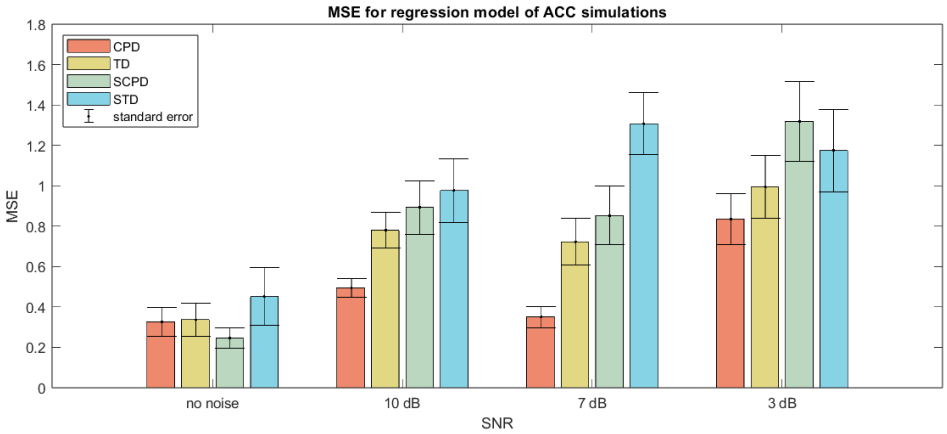

Figure 5.17: Results from applying the regression algorithms to the ACC simulations. The mean square error is calculated using predicted outputs from the regression models and the real simulated outputs using 5-fold cross-validation.

in Fig. 5.18. For both the MMN and ACC simulations, the regression model using the CPD performs the best in most cases. Using the TD instead of the CPD results in the same performance at best but often in a higher MSE. The sparsity constraint does only improve the accuracy of the model in a few cases, but in general the algorithms perform worse with this extra constraint.

5.3.2. Feature Extraction

Just like for the classification algorithms, the regression algorithms provide the possibility to find underlying mechanisms on which the decisions are made. In both the CPD and TD-based regression algorithms, factor matrices are created to correspond to each mode of the tensor. These matrices contain the weights corresponding to the samples of each mode and therefore can tell something of the importance of location or time samples in computing the estimated output.

These factor matrices are displayed in Fig. 5.19 and 5.20 for the MMN and ACC simulation, respectively, with an SNR of 7 dB. The regression algorithms that perform the best, CPD and TD, show the largest peak at around 125 ms, where the peaks relevant to the output can be found. In terms of location and stimulus, no such strong signs can be seen that the regression weights correspond to what is expected from the simulation. The sparse constraint results in weights for the time mode with only a few nonzero matrix entries, which approximately correspond to the expected locations. In general the sparsity constraint results in less useful features.

When inspecting the weights of regression on the ACC simulation, the main thing to notice is the that both CPD and TD indicate the times where the ACC peaks happen. Especially in the TD case the peaks

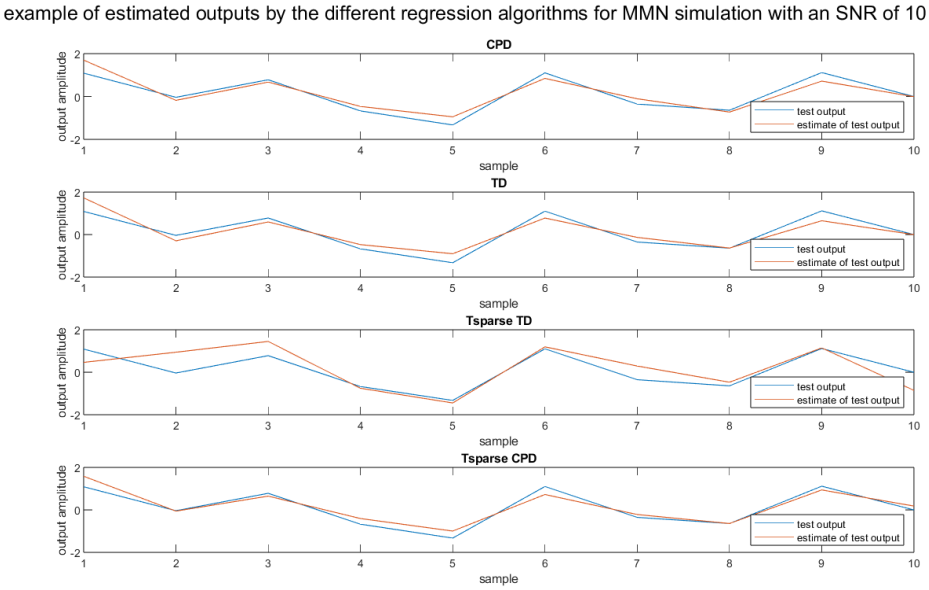

Figure 5.18: An example of what the estimate test output looks like next to the real test output.

are clearly visible around 120 and 620 ms, but only the second peak should be considered relevant for the regression output. The sparsity constraint does not make these peaks more clear, just like for the MMN simulation.

5.4. Discussion

The discriminant analysis algorithms can help distinguish between different classes. HODA appears to be more suitable for ERPs of a higher quality, while SHODA and BSHODA show better classification rates for a lower SNR. This indicates that these tensors can distinguish times and locations that are different between classes from samples that are not of importance. When analyzing the resulting projection matrices, it is nevertheless not possible to identify these samples reliably. It will still be interesting to look at the projection matrices, but it is not expected that new insights will be provided by this. Only the projection matrix related to the different segments, that is obtained by applying HODA to the segmented tensor, shows really clearly where the classes differ.

Trying to find a reliable relation between an input tensor and an output using the described tensor regression methods does work for lower higher SNR. The regression algorithm based on the CPD works best in general. The weight matrices that are found by the regression algorithms do not show any relevant information.

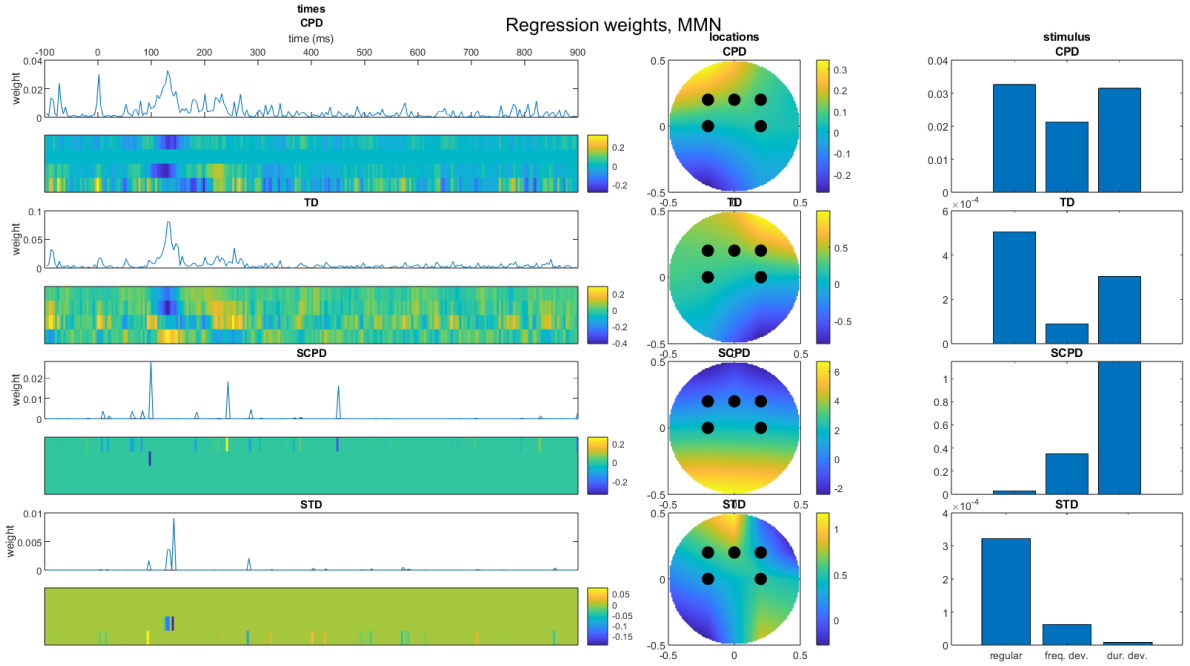

Figure 5.19: Weights of features from applying regression to MMN simulation. The projection matrices corresponding to time are displayed as heatmaps, with the squared sum of the R columns above. The weights from the projection matrices corresponding to the location are displayed in the head plots, and the ERPs produced by the different stimuli are in the bar graphs.

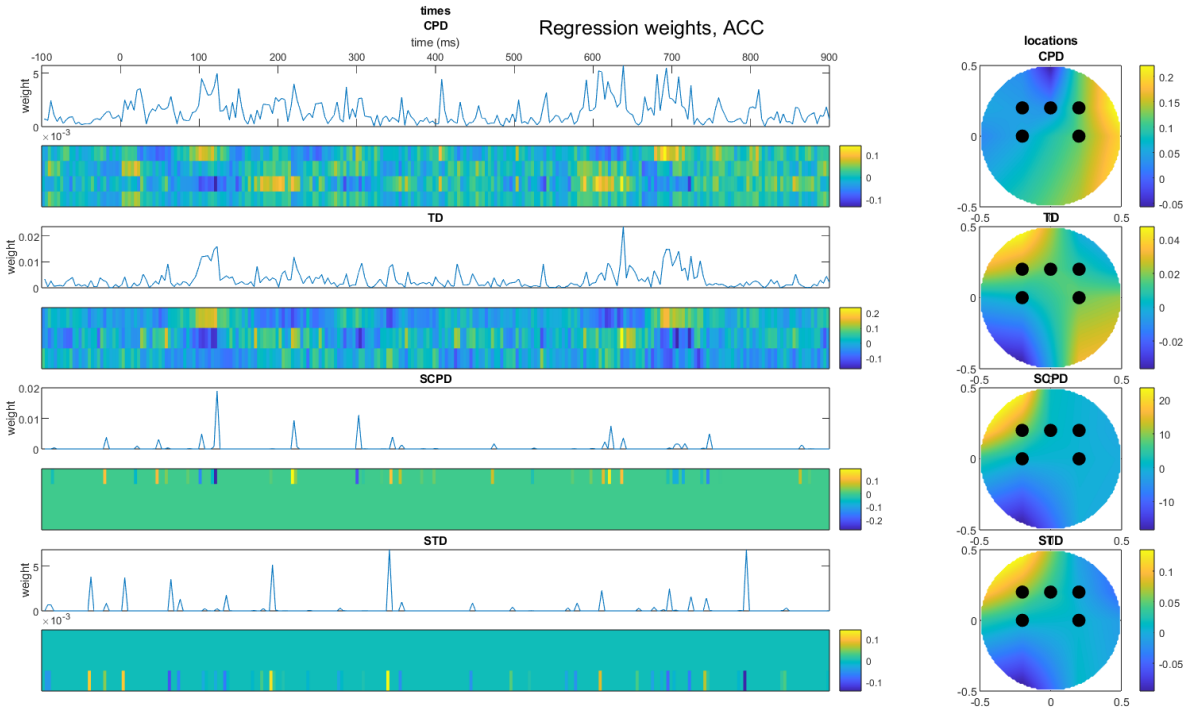

Figure 5.20: Weights of features from applying regression to ACC simulation. The projection matrices corresponding to time are displayed as heatmaps, with the squared sum of the R columns above. The weights from the projection matrices corresponding to the location are displayed in the head plots.

6

Results from KHL data

In this chapter, the ERPs from Chapter 3 and the algorithm from Chapter 4 are combined to try and gain new insights on the data from the KHL. Because of the very low number of successful ACC trials, only the results obtained from the MMN measurements are discussed in this chapter. The accuracy of the DA methods will be shown and discussed, and for the regression results some short comments will be made. The only method to find new insights about the KHL data is by analysing the projection matrix of the segmented tensor, because this was the only method that showed time samples that were relevant according to the simulations.

6.1. Discriminant Analysis

Discriminant analysis is performed on subjects belonging to different classes. Each disorder that was measured can be considered such a class. This does, however, result in a low number of subjects per class. To solve this disorder, it can be grouped in classes as well. One of the groupings that is used is the three disorders that are considered to have no effect on the MMN and ACC. This class is used together with classes for each of the other disorders in DA. Another grouping of classes is to compare GRIN/GRIA to all other disorders combined in a group because the results from Chapter 3 show that this disorder deviates the most from the others. In the case of the MMN measurements, the two best ERPs are from GRIN/GRIA and Sturge Weber syndrome, so these are also compared to each other. An overview of these classes that are considered for discriminant analysis is shown in Fig. 6.1

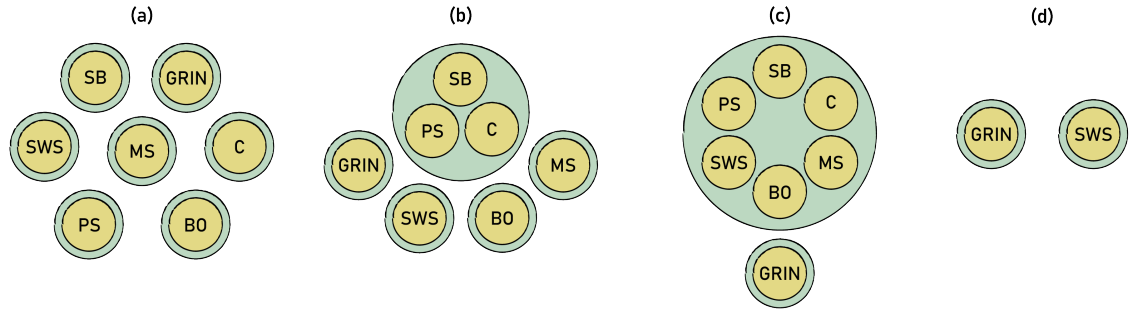

Figure 6.1: Different combinations of disorders are used to make different classes. (a) All disorders are considered a separate class. (b) The disorders which are expected to have no effect on the MMN waveform are used as a “control group” class. (c) The disorder that stood out the most in the analysis of the ERP waveforms, GRIN/GRIA, is treated as a separate class from all other disorders. (d) For the analysis of the MMN measurements, GRIN/GRIA and Sturge Weber are considered different classes.

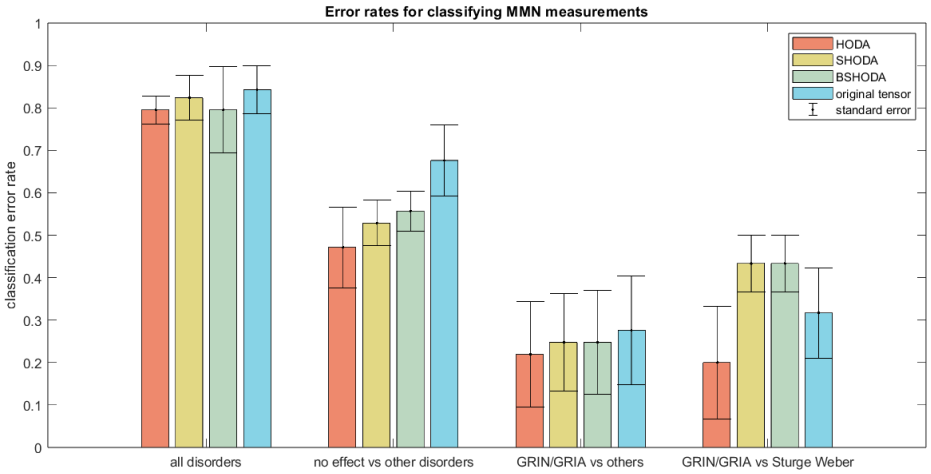

Figure 6.2: Classification errors for the different DA methods on the MMN data, by using 5-fold cross-validation and 1 nearest neighbour classification. DA is used on different groupings of disorders as classes.

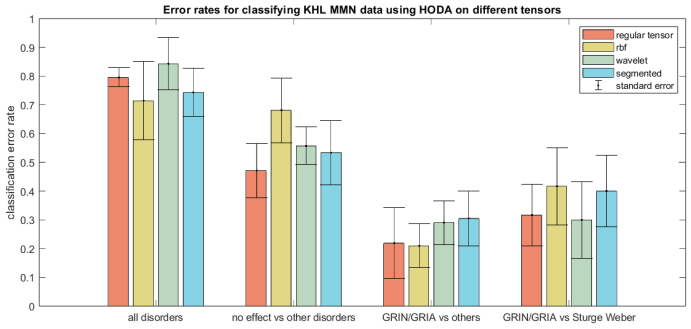

Figure 6.3: Classification errors when HODA and classification are applied to regular and extended tensors.

6.1.1. Results

The same methods that are used to analyse the discriminant methods for the simulated data are used for the KHL data as well. The only difference is that no different groups are made because the data is limited. This, unfortunately, results in higher standard errors. The resulting classification rates are shown in Fig. 6.2. It is important to note that the lower classification rates for different groupings of classes are due to the lower number of classes the algorithm has to choose from, and not that the performance is better on these groups. For all disorders as a separate class and GRIN/GRIA versus the others, the discriminant analysis methods improve on classification rate but not significantly. When three classes are considered as a control group, the DA methods do result in a better classification rate.

To see if any nonlinearities or frequency bands differentiate the different classes, HODA is also applied to the altered tensors described in Chapter 4. The results from this can be seen in Fig. 6.3, which shows that no significant improvements are made in comparison to using the regular tensor.

6.1.2. Features Extracted

In Chapter 5, the features that could be extracted from the projection matrices are not very useful. Only when the tensor is segmented can the segments indicate which groups of time samples are of most importance when differentiating between the various classes. For the different groupings of disorders the weights of the segments are shown in Fig. 6.4, 6.5, 6.6 and 6.7. All these figures, except the last, indicate that time segments are important within the range that is expected. Fig. 6.4 does not show any very large peaks, but the main difference between all classes happens around 300 ms. Fig. 6.5 shows that the difference between the control classes and the other disorders also happens around 300 ms.

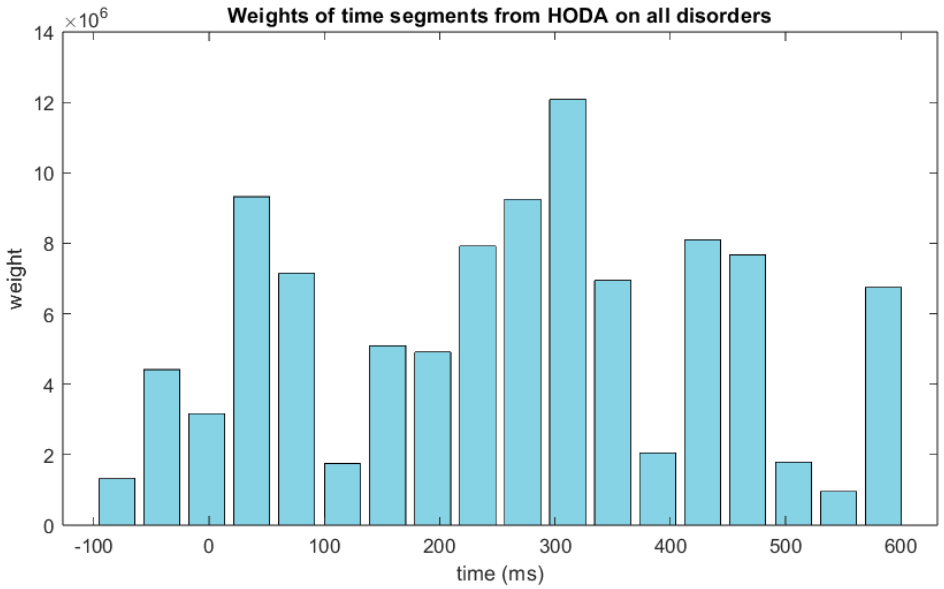

Figure 6.4: Weights of each time segment, from the projection matrix corresponding to the segments found by HODA.

Fig. 6.6 indicates that GRIN/GRIA differs from the other classes a bit later, around 400 ms. This is a bit surprising since visual inspection shows that this disorder has a higher negative peak around 200 ms as the most obvious difference with the other disorders. The last figure (Fig. 6.7) Does not give useful information since it indicates the main difference between GRIN/GRIA and SWS occurs before the stimulus is presented. When looking at the MMN waveforms, this might be explained by a small peak in the GRIN/GRIA MMN around 20 ms before the stimulus.

6.1.3. Discussion

The discriminant analysis algorithm does improve on the classification rate a bit, but this classification rate is, in all cases, far too high to be considered useful. With better machine learning algorithms, the classification rate might become better, and the discriminant analysis is still a useful tool for preprocessing the data.

When inspecting the projection matrices related to time segments, some interesting observations can be made. These can be used to take a second look at the ERP and MMN waveforms with these observations in mind. Considering the classification rates and visual inspection of these projection matrices, this should not be used to make any hard conclusions on when the differences between the classes occur.

6.2. Tensor Regression

The children that are measured in the KHL also undergo other tests. One of these tests is their ability to hear sounds while noise is played. These test results in scores that can be related to the measurements using tensor regression. This section will explore if the tensor regression methods from Chapter 4 can predict these test scores and can relate certain samples from the input tensor to the scores. Of the successful MMN measurements, 12 children have test scores, and for the ACC measurements, it is 7 children. The test scores that are used are digit-in-noise (DIN) scores, which is an increasingly popular method to quantify hearing loss [95].

Unfortunately, the tensor regression methods were not able to generate models that could predict these test scores. The “best” MSEs between predictions and test scores are obtained by having the regularization parameter set to a value that results in the prediction always being very close to zero. A lower regularization parameter results in overfitting to the subjects used to fit the model. This could be the case due to the low number of children with successful measurements and test scores or the lack of a

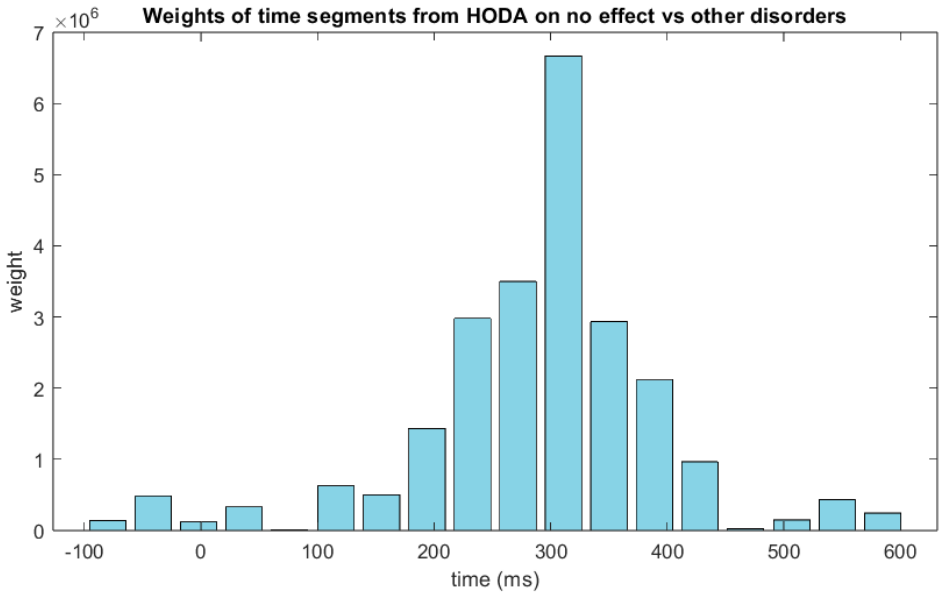

Figure 6.5: Weights of each time segment, from the projection matrix corresponding to the segments found by HODA.

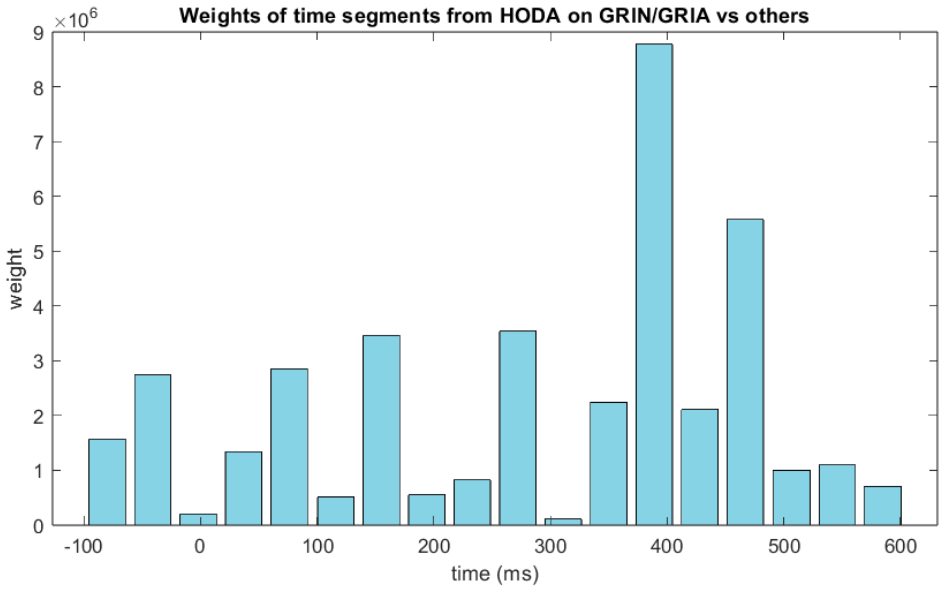

Figure 6.6: Weights of each time segment, from the projection matrix corresponding to the segments found by HODA.

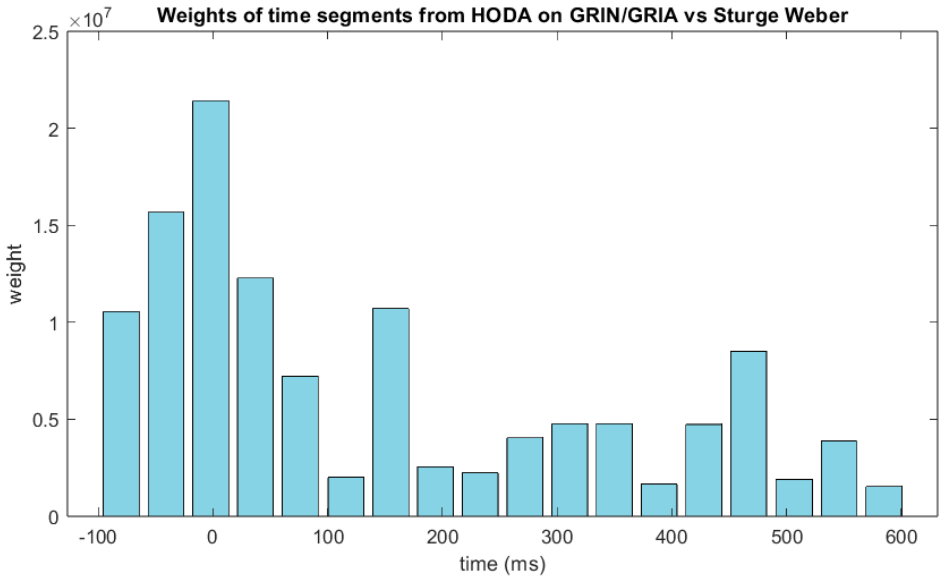

Figure 6.7: Weights of each time segment, from the projection matrix corresponding to the segments found by HODA.

strong relation between the test scores and measurements, but likely due to both.

7

Discussion and Future Work

Working with the data from the Kinderhersenlab provides some interesting challenges. A relatively low number of successful measurements is obtained, and the quality of the measurements that are used is relatively low.

Because of the tensor structure of the data, it can be beneficial to use methods that can be applied directly on tensors. These measurements can be combined with other information from the KHL, mainly the disorders and various test scores. The disorders can be viewed as different classes, making it interesting to try and optimize classification. Doing this using discriminant analysis instead of complex machine learning models also gives insight into the underlying mechanisms of how the algorithm tries to differentiate the classes. This insight can be extracted from the projection matrices produced by higher-order discriminant analysis. The test scores from the KHL are continuous variables and can, therefore, be used in combination with the measurements in a regression model. Tensor networks can be used to reduce the amount of weights that have to be estimated in this model.

Due to the limited number of subjects available, both discriminant analysis and regression can suffer from over-fitting or result in a locally optimal solution. Therefore, the methods found in the literature are altered in this thesis to include constraints tailored to the specific data. These constraints are based on the principle that the ERPs of the different disorders are likely to be different in certain components and not in the whole waveform. This is implemented by applying a relaxed cardinality constraint, the l-1 norm, to the projection matrices in HODA and to the factor matrices of the weights in regression. Because each of these matrices is related to only one mode of the input tensor, this constraint can be set to the matrices related to time. This promotes only sparsity in time, where it is expected based on the nature of the data.

Simulations show that applying higher-order discriminant analysis before using the data in a 1-nearest neighbour classification algorithm increases the classification rate. This suggests that the algorithm is doing what it is supposed to: project the tensor to a space in which the different classes are more separated from each other. The additional constraint does, in most cases, decrease the classification rate compared to HODA, meaning that the constrained solution is limited to a point where the optimal solution is no longer found. When more noise is added, the performance gap between HODA and (B)SHODA becomes smaller, and in one ACC simulation, the latter algorithms even perform better. This suggests that the sparsity constraint might have some use in specific noisy cases.

When applying tensor regression to the simulations, a suitable model can be found in the case of a high SNR. The regression works best when it is based on the CPD, but the TD also performs nearly as well in most cases. Just like for HODA, adding sparsity constraints does worsen the performance in most cases.

When applying the HODA algorithms to the real data, the classification rate is improved in some cases.

This is dependent on which types of classes are constructed using the various disorders. Classification rates are, however, far too high to be useful in any way. When the tensor containing the measurements is altered to be segmented in time before applying HODA, certain windows in time are highlighted as more important. These time segments are both within and outside the time windows where the disorders are expected to differ. Although the projection matrices do not guarantee what are the most or only important time segments, they can be used to examine the ERP waveforms in a new light.

Regression can not be used to reliably predict test scores. This is likely due to the low number of subjects that can be used to find a suitable model.

7.1. Future Work

The measurements from the KHL produce a grand average that is consistent with the literature for both the MMN and ACC. The more interesting question is whether differences between the disorders that are measured can be seen. With the current amount of subjects that are measured, this is not clearly the case. The only disorder that stands out quite a bit is GRIN/GRIA. When more data is available in the future, a new look should be taken at the resulting ERP plots to get more reliable results.

The algorithms that are used to try and find some more interesting results from the data show reasonable results when applied to simulated data, but these results can be improved. For both the tensor discriminant analysis and regression, algorithms are used from the literature that is not the most state-of-the-art anymore. These basic algorithms were chosen, however, because it is less complicated to tailor them to perform better on ERP data. When optimizing classification and regression performance, other more recently developed algorithms might work better.

While the reasoning for the usefulness of sparsity constraints might be valid, in practice, this addition to the algorithms does not improve the performance in most cases. It is still interesting that these alternative approaches do, in some cases, improve the algorithms. Other implementations of these sparsity constraints or slightly different constraint might therefore work in some cases as well.

When dealing with larger tensors, the faster approach to HODA in the methodology section does speed up the algorithm by quite a bit. This is, however, very dependent on the input data and the hyperparameters that are chosen, as well as the specific implementation Matlab uses for certain operations. Mathematical proof that this algorithm is faster still has to be found.

Using extended versions of the input tensor using the radial basis function and wavelet transform still has more potential. Very basic versions of these functions are used in this thesis, while other versions might work better. the σ in rbf can be tuned for optimal performance, instead of being simply set to one. For the Wavelet transform there is an even greater variety of options instead of Matlabs default cwt implementation. Different wavelets can be used, as well as other frequency bands.

Due to the time limitations of this project, the altered input tensors are only applied to the HODA algorithm. These inputs can however also be used on the regression algorithms in the future.

The proposed algorithms were applied to the KHL data in the hope of uncovering new insights into the disorders. However, the features resulting from both the DA and regression algorithms do not give clear hints as to what the main differences between the disorders are. For discriminant analysis, however, this is a little better than in the case of regression. The results will hopefully be better by using more subjects in the future.

8

Conclusion

In conclusion, working with the data from the Kinderhersenlab presented some interesting challenges, including a relatively low number of successful measurements and low measurement quality. The tensor structure of the data provided an opportunity to use methods directly applicable to tensors, and combining these measurements with other information from the KHL, such as disorders and test scores, opened possibilities for classification optimization and regression modelling.

The limitations of subjects available for the study might be countered by having the algorithms tailored with constraints specific to the data to address overfitting and locally optimal solutions. Simulations provided insights into the effectiveness of higher-order discriminant analysis and regression algorithms, and results indicated the potential for sparsity constraints in specific noisy scenarios. However, challenges were encountered in applying regression algorithms to the simulated data, highlighting areas for future exploration and improvement.

Moving forward, future work with the KHL measurements will benefit from an increased subject pool, enabling a more comprehensive analysis of the differences between the measured disorders. Furthermore, utilizing more state-of-the-art algorithms for tensor discriminant analysis and regression, specifically tailored for ERP data, could lead to improved results.

Overall, the methodologies used in this study have the potential for further exploration and refinement, particularly with the prospect of more comprehensive data and advancements in algorithmic approaches.

Bibliography

[1] H. Berger, “Über das elektroenkephalogramm des menschen,” *Archiv für psychiatrie und nervenkrankheiten*, vol. 87, no. 1, pp. 527–570, 1929.

[2] P. Korpilahti and H. Lang, “Auditory erp components and mismatch negativity in dysphasic children,” *Electroencephalography and Clinical Neurophysiology*, vol. 91, no. 4, pp. 256–264, 1994. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0013469494901899

[3] R. Näätänen, T. Kujala, C. Escera, T. Baldeweg, K. Kreegipuu, S. Carlson, and C. Ponton, “The mismatch negativity (mmn) – a unique window to disturbed central auditory processing in ageing and different clinical conditions,” *Clinical Neurophysiology*, vol. 123, no. 3, pp. 424–458, 2012. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1388245711006882

[4] R. Näätänen, J. Todd, and U. Schall, “Mismatch negativity (mmn) as biomarker predicting psychosis in clinically at-risk individuals,” *Biological Psychology*, vol. 116, pp. 36–40, 2016, understanding the neurobiology of MMN and its reduction in schizophrenia. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0301051115300715

[5] J.-R. Kim, “Acoustic change complex: Clinical implications,” *J. Audiol. Otol.*, vol. 19, no. 3, pp. 120–124, Dec. 2015.

[6] K. McGuire, G. M. Firestone, N. Zhang, and F. Zhang, “The acoustic change complex in response to frequency changes and its correlation to cochlear implant speech outcomes,” *Frontiers in Human Neuroscience*, vol. 15, 2021. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fnhum.2021.757254

[7] S. J. Luck, *An introduction to the event-related potential technique*. MIT press, June 2014.

[8] R. Näätänen, A. Gaillard, and S. Mäntysalo, “Early selective-attention effect on evoked potential reinterpreted,” *Acta Psychologica*, vol. 42, no. 4, pp. 313–329, 1978. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0001691878900069

[9] K. Fitzgerald and J. Todd, “Making sense of mismatch negativity,” *Frontiers in Psychiatry*, vol. 11, p. 468, 2020.

[10] I. Winkler, “Interpreting the mismatch negativity,” *Journal of Psychophysiology*, vol. 21, no. 3-4, pp. 147–163, 2007. [Online]. Available: https://doi.org/10.1027/0269-8803.21.34.147

[11] Y. I. Fishman, “The mechanisms and meaning of the mismatch negativity,” *Brain Topography*, vol. 27, no. 4, pp. 500–526, Jul 2014. [Online]. Available: https://doi.org/10.1007/s10548-013-0337-3

[12] S. Pakarinen, R. Takegata, T. Rinne, M. Huotilainen, and R. Näätänen, “Measurement of extensive auditory discrimination profiles using the mismatch negativity (mmn) of the auditory event-related potential (erp),” *Clinical Neurophysiology*, vol. 118, no. 1, pp. 177–185, 2007. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1388245706014301

[13] R. Näätänen, E. S. Sussman, D. Salisbury, and V. L. Shafer, “Mismatch negativity (mmn) as an index of cognitive dysfunction,” *Brain Topography*, vol. 27, no. 4, pp. 451–466, Jul 2014. [Online]. Available: https://doi.org/10.1007/s10548-014-0374-6

[14] J. M. Ostroff, B. A. Martin, and A. Boothroyd, “Cortical evoked response to acoustic change within a syllable,” *Ear and Hearing*, vol. 19, no. 4, 1998. [Online]. Available: https://journals.lww.com/ear-hearing/fulltext/1998/08000/cortical_evoked_response_to_acoustic_change_within.4.aspx

[15] B. A. Martin and A. Boothroyd, “Cortical, auditory, evoked potentials in response to changes of spectrum and amplitude,” *The Journal of the Acoustical Society of America*, vol. 107, no. 4, pp. 2155–2161, 04 2000. [Online]. Available: https://doi.org/10.1121/1.428556

[16] ——, “Cortical, auditory, event-related potentials in response to periodic and aperiodic stimuli with the same spectral envelope,” *Ear and Hearing*, vol. 20, no. 1, 1999. [Online]. Available: https://journals.lww.com/ear-hearing/fulltext/1999/02000/cortical,_auditory,_event_related_potentials_in.4.aspx

[17] K. L. Tremblay, L. Friesen, B. A. Martin, and R. Wright, “Test-retest reliability of cortical evoked potentials using naturally produced speech sounds,” *Ear and Hearing*, vol. 24, no. 3, 2003. [Online]. Available: https://journals.lww.com/ear-hearing/fulltext/2003/06000/test_retest_reliability_of_cortical_evoked.6.aspx

[18] A. S. Martinez, L. S. Eisenberg, and A. Boothroyd, “The acoustic change complex in young children with hearing loss: A preliminary study,” *Semin Hear*, vol. 34, no. 4, pp. 278–287, 2013.

[19] S. A. Small and J. F. Werker, “Does the acc have potential as an index of early speech discrimination ability? a preliminary study in 4-month-old infants with normal hearing,” *Ear and Hearing*, vol. 33, no. 6, 2012. [Online]. Available: https://journals.lww.com/ear-hearing/fulltext/2012/11000/does_the_acc_have_potential_as_an_index_of_early.11.aspx

[20] S. van Mierlo, L. de Jong, and M. Adank, “Protocol auditieive event-related potential (erp) metingen in het kinderhersenlab,” Apr. 2023.

[21] A. J. Copp, N. S. Adzick, L. S. Chitty, J. M. Fletcher, G. N. Holmbeck, and G. M. Shaw, “Spina bifida,” *Nature reviews Disease primers*, vol. 1, no. 1, pp. 1–18, 2015.

[22] B. Lindquist, H. Jacobsson, M. Strinnholm, and M. Peny-Dahlstrand, “A scoping review of cognition in spina bifida and its consequences for activity and participation throughout life,” *Acta paediatrica*, vol. 111, no. 9, pp. 1682–1694, 2022.

[23] S. Endele, G. Rosenberger, K. Geider, B. Popp, C. Tamer, I. Stefanova, M. Milh, F. Kortüm, A. Fritsch, F. K. Pientka *et al.*, “Mutations in grin2a and grin2b encoding regulatory subunits of nmda receptors cause variable neurodevelopmental phenotypes,” *Nature genetics*, vol. 42, no. 11, pp. 1021–1026, 2010.

[24] Sep 2023. [Online]. Available: https://grinsyndroom.nl/

[25] D. L. Da Silva, F. Palheta Neto, S. G. Carneiro, A. C. P. Palheta, M. Monteiro, S. C. Cunha *et al.*, “Crouzon’s syndrome: literature review,” *Intl Arch Otorhinolaryngol*, vol. 12, no. 3, pp. 436–441, 2008.

[26] K. A. Thomas-Sohl, D. F. Vaslow, and B. L. Maria, “Sturge-weber syndrome: a review,” *Pediatric neurology*, vol. 30, no. 5, pp. 303–310, 2004.

[27] M. Zallmann, R. J. Leventer, M. T. Mackay, M. Ditchfield, P. S. Bekhor, and J. C. Su, “Screening for sturge-weber syndrome: A state-of-the-art review,” *Pediatric dermatology*, vol. 35, no. 1, pp. 30–42, 2018.

[28] R. Dobson and G. Giovannoni, “Multiple sclerosis–a review,” *European journal of neurology*, vol. 26, no. 1, pp. 27–40, 2019.

[29] J. Rusz, B. Benova, H. Ruzickova, M. Novotny, T. Tykalova, J. Hlavnicka, T. Uher, M. Vaneckova, M. Andelova, K. Novotna *et al.*, “Characteristics of motor speech phenotypes in multiple sclerosis,” *Multiple sclerosis and related disorders*, vol. 19, pp. 62–69, 2018.

[30] J. Jung, D. Morlet, B. Mercier, C. Confavreux, and C. Fischer, “Mismatch negativity (mmn) in multiple sclerosis: an event-related potentials study in 46 patients,” *Clinical neurophysiology*, vol. 117, no. 1, pp. 85–93, 2006.

[31] N. E. Neef, A. Anwander, and A. D. Friederici, “The neurobiological grounding of persistent stuttering: from structure to function,” *Current neurology and neuroscience reports*, vol. 15, pp. 1–11, 2015.

[32] E. Jansson-Verkasalo, K. Eggers, A. Järvenpää, K. Suominen, B. Van den Bergh, L. De Nil, and T. Kujala, “Atypical central auditory speech-sound discrimination in children who stutter as indexed by the mismatch negativity,” *Journal of fluency disorders*, vol. 41, pp. 1–11, 2014.

[33] F. Brioude, A. Toutain, E. Giabicani, E. Cottereau, V. Cormier-Daire, and I. Netchine, “Overgrowth syndromes—clinical and molecular aspects and tumour risk,” *Nature Reviews Endocrinology*, vol. 15, no. 5, pp. 299–311, 2019.

[34] G. López-Arango, F. Deguire, K. Agbogba, M.-A. Boucher, I. S. Knoth, R. El-Jalbout, V. Côté, A. Damphousse, S. Kadoury, and S. Lippé, “Impact of brain overgrowth on sensorial learning processing during the first year of life,” *Frontiers in human neuroscience*, vol. 16, p. 928543, 2022.

[35] L. Pion-Tonachini, K. Kreutz-Delgado, and S. Makeig, “ICLabel: An automated electroencephalographic independent component classifier, dataset, and website,” *Neuroimage*, vol. 198, pp. 181–197, May 2019.

[36] A. Delorme and S. Makeig, “Eeglab: An open source toolbox for analysis of single-trial eeg dynamics including independent component analysis,” *Journal of Neuroscience Methods*, vol. 134, no. 1, pp. 9–21, 2004. [Online]. Available: 10.1016/j.jneumeth.2003.10.009

[37] J. Lopez-Calderdon and S. Luck, “Erplab: An open-source toolbox for the analysis of event-related potentials,” *Frontiers in human neuroscience*, 2014. [Online]. Available: https://doi.org/10.3389/fnhum.2014.00213

[38] E. B. Montgomery, “Noise and Artifact,” in *Intraoperative Neurophysiological Monitoring for Deep Brain Stimulation: Principles, Practice and Cases*. Oxford University Press, 07 2014. [Online]. Available: https://doi.org/10.1093/med/9780199351008.003.0006

[39] J. A. Urigüen and B. Garcia-Zapirain, “Eeg artifact removal—state-of-the-art and guidelines,” *Journal of Neural Engineering*, vol. 12, no. 3, April 2015.

[40] W. Mumtaz, S. Rasheed, and A. Irfan, “Review of challenges associated with the eeg artifact removal methods,” *Biomedical Signal Processing and Control*, vol. 68, p. 102741, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1746809421003384

[41] S. Makeig, A. Bell, T.-P. Jung, and T. J. Sejnowski, “Independent component analysis of electroencephalographic data,” in *Advances in Neural Information Processing Systems*, D. Touretzky, M. Mozer, and M. Hasselmo, Eds., vol. 8. MIT Press, 1995. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/1995/file/754dda4b1ba34c6fa89716b85d68532b-Paper.pdf

[42] B. A. Pramudita, Y. D. Nugroho, I. Ardiyanto, M. I. Shapiai, and N. A. Setiawan, “Removing ocular artefacts in eeg signals by using combination of complete eemd (ceemd) — independent component analysis (ica) based outlier data,” in *2017 International Conference on Robotics, Automation and Sciences (ICORAS)*, Nov 2017, pp. 1–5.

[43] N. P. Castellanos and V. A. Makarov, “Recovering eeg brain signals: Artifact suppression with wavelet enhanced independent component analysis,” *Journal of Neuroscience Methods*, vol. 158, no. 2, pp. 300–312, 2006. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0165027006002834

[44] S. Sur and V. K. Sinha, “Event-related potential: An overview,” *Ind Psychiatry J*, vol. 18, no. 1, pp. 70–73, Jan. 2009.

[45] C. J. Patrick, E. M. Bernat, S. M. Malone, W. G. Iacono, R. F. Krueger, and M. McGue, “P300 amplitude as an indicator of externalizing in adolescent males,” *Psychophysiology*, vol. 43, no. 1, pp. 84–92, Jan. 2006.

[46] E. Bramon, S. Rabe-Hesketh, P. Sham, R. M. Murray, and S. Frangou, “Meta-analysis of the P300 and P50 waveforms in schizophrenia,” *Schizophr Res*, vol. 70, no. 2-3, pp. 315–329, Oct. 2004.

[47] B. A. Clementz, M. A. Geyer, and D. L. Braff, “P50 suppression among schizophrenia and normal comparison subjects: A methodological analysis,” *Biological Psychiatry*, vol. 41,

no. 10, pp. 1035–1044, 1997. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0006322396002089

[48] B. O'Donnell, J. Vohs, W. Hetrick, C. Carroll, and A. Shekhar, "Auditory event-related potential abnormalities in bipolar disorder and schizophrenia," *International Journal of Psychophysiology*, vol. 53, no. 1, pp. 45–55, 2004. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167876004000169

[49] K. K. Schulze, M.-H. Hall, C. McDonald, N. Marshall, M. Walshe, R. M. Murray, and E. Bramon, "P50 auditory evoked potential suppression in bipolar disorder patients with psychotic features and their unaffected relatives," *Biological Psychiatry*, vol. 62, no. 2, pp. 121–128, 2007, bipolar Disorder: Neurocircuitry Neurodevelopment. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0006322306010067

[50] M. Hansenne, W. Pitchot, A. Gonzalez Moreno, I. Urcelay Zaldua, and M. Ansseau, "Suicidal behavior in depressive disorder: An event-related potential study," *Biological Psychiatry*, vol. 40, no. 2, pp. 116–122, 1996. [Online]. Available: https://www.sciencedirect.com/science/article/pii/000632239500372X

[51] S. Sun, C. Zhang, and Y. Lu, "The random electrode selection ensemble for eeg signal classification," *Pattern Recognition*, vol. 41, no. 5, pp. 1663–1675, 2008. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0031320307004864

[52] R. Palaniappan and K. Ravi, "Improving visual evoked potential feature classification for person recognition using pca and normalization," *Pattern Recognition Letters*, vol. 27, no. 7, pp. 726–733, 2006. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S016786550500320X

[53] J. Dien, "Applying principal components analysis to event-related potentials: A tutorial," *Developmental Neuropsychology*, vol. 37, no. 6, pp. 497–517, 2012, pMID: 22889342. [Online]. Available: https://doi.org/10.1080/87565641.2012.697503

[54] D. J. Krusienski1, E. W. Sellers, F. Cabestaing, S. Bayoudh, D. J. McFarland, T. M. Vaughan, and J. R. Wolpaw, "A comparison of classification techniques for the p300 speller," *Journal of Neural Engineering*, vol. 3, no. 4, October 2006. [Online]. Available: DOI10.1088/1741-2560/3/4/007

[55] B. Blankertz, M. T. S. Lemm, S. Haufe, and K. Mller, "Single-trial analysis and classification of erp components—a tutorial," *NeuroImage*, vol. 56, no. 2, pp. 814–825, 2011.

[56] A. Sharma and K. K. Paliwal, "Linear discriminant analysis for the small sample size problem: an overview," *International Journal of Machine Learning and Cybernetics*, vol. 6, no. 3, pp. 443–454, Jun 2015. [Online]. Available: https://doi.org/10.1007/s13042-013-0226-9

[57] V. Peterson, H. L. Rufiner, and R. D. Spies, "Generalized sparse discriminant analysis for event-related potential classification," *Biomedical Signal Processing and Control*, vol. 35, pp. 70–78, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1746809417300514

[58] M.-P. Hosseini, A. Hosseini, and K. Ahi, "A review on machine learning for eeg signal processing in bioengineering," *IEEE Reviews in Biomedical Engineering*, vol. 14, pp. 204–218, 2021.

[59] A. Bablani, D. R. Edla, and S. Dodia, "Classification of eeg data using k-nearest neighbor approach for concealed information test," *Procedia Computer Science*, vol. 143, pp. 242–249, 2018, 8th International Conference on Advances in Computing Communications (ICACC-2018). [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050918320891

[60] S. Bhaduri, A. Khasnobish, R. Bose, and D. N. Tibarewala, "Classification of lower limb motor imagery using k nearest neighbor and naïve-bayesian classifier," in *2016 3rd International Conference on Recent Advances in Information Technology (RAIT)*, 2016, pp. 499–504.

[61] N. Saini, S. Bhardwaj, R. Agarwal, and S. Chandra, "Information detection in brain using wavelet features and k-nearest neighbor," in *2021 6th International Conference on Communication and Electronics Systems (ICCES)*, 2021, pp. 1704–1709.

[62] J. Yang, W. Li, S. Wang, J. Lu, and L. Zou, "Classification of children with attention deficit hyperactivity disorder using pca and k-nearest neighbors during interference control task," in *Advances in*

Cognitive Neurodynamics (V), R. Wang and X. Pan, Eds. Singapore: Springer Singapore, 2016, pp. 447–453.

[63] A. Sharmila and P. Geethanjali, "Dwt based detection of epileptic seizure from eeg signals using naive bayes and k-nn classifiers," *IEEE Access*, vol. 4, pp. 7716–7727, 2016.

[64] F. Cong, Q.-H. Lin, L.-D. Kuang, X.-F. Gong, P. Astikainen, and T. Ristaniemi, "Tensor decomposition of eeg signals: A brief review," *Journal of Neuroscience Methods*, vol. 248, pp. 59–69, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0165027015001016

[65] Q. Zhao, C. F. Caiafa, D. Mandic, L. Zhang, T. Ball, A. Schulze-bonhage, and A. Cichocki, "Multilinear subspace regression: An orthogonal tensor decomposition approach," in *Advances in Neural Information Processing Systems*, J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Weinberger, Eds., vol. 24. Curran Associates, Inc., 2011. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2011/file/1343777b8ead1cef5a79b78a1a48d805-Paper.pdf

[66] Z. S. Bonab and M. B. Shamsollahi, "Event related potentials extraction using low-rank tensor decomposition," in *2022 30th International Conference on Electrical Engineering (ICEE)*, 2022, pp. 931–935. [Online]. Available: https://doi.10.1109/ICEE55646.2022.9827218

[67] N. D. Sidiropoulos, L. De Lathauwer, X. Fu, K. Huang, E. E. Papalexakis, and C. Faloutsos, "Tensor decomposition for signal processing and machine learning," *IEEE Transactions on Signal Processing*, vol. 65, no. 13, pp. 3551–3582, 2017.

[68] F. Cong, A.-H. Phan, P. Astikianen, Q. Zhao, Q. Wu, J. K. Hietanen, T. Ristaniemi, and A. Cichocki, "Multi-domain feature extraction for small event-related potentials through nonnegative multi-way array decomposition from low dense array eeg," *International Journal of Neural Systems*, vol. 23, no. 02, p. 1350006, 2013, pMID: 23578056. [Online]. Available: https://doi.org/10.1142/S0129065713500068

[69] Y. Zhang, G. Zhou, Q. Zhao, J. Jin, X. Wang, and A. Cichocki, "Spatial-temporal discriminant analysis for erp-based brain-computer interface," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 21, no. 2, pp. 233–243, 2013.

[70] A. Kumar, E. Pirogova, and J. Q. Fang, "Classification of error-related potentials using linear discriminant analysis," in *2018 IEEE-EMBS Conference on Biomedical Engineering and Sciences (IECBES)*, 2018, pp. 18–21.

[71] B. Blankertz, S. Lemm, M. Treder, S. Haufe, and K.-R. Müller, "Single-trial analysis and classification of erp components — a tutorial," *NeuroImage*, vol. 56, no. 2, pp. 814–825, 2011, multivariate Decoding and Brain Reading. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1053811910009067

[72] A. Mueller, G. Candrian, J. D. Kropotov, V. A. Ponomarev, and G.-M. Baschera, "Classification of adhd patients on the basis of independent erp components using a machine learning system," *Nonlinear Biomedical Physics*, vol. 4, no. 1, p. S1, Jun 2010. [Online]. Available: https://doi.org/10.1186/1753-4631-4-S1-S1

[73] I. Winkler, S. Brandl, F. Horn, E. Waldburger, C. Allefeld, and M. Tangermann, "Robust artifactual independent component classification for bci practitioners," *Journal of Neural Engineering*, vol. 11, no. 3, p. 035013, may 2014. [Online]. Available: https://dx.doi.org/10.1088/1741-2560/11/3/035013

[74] S. Yan, D. Xu, Q. Yang, L. Zhang, X. Tang, and H.-J. Zhang, "Discriminant analysis with tensor representation," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, 2005, pp. 526–532 vol. 1.

[75] G. Quanxue, H. Jun, H. Xiujuan, and L. Yiying, "Directional multimode subspace analysis with tensor representation-discriminant feature extraction," in *2010 International Conference on Computational Aspects of Social Networks*, 2010, pp. 361–364.

[76] Q. Li and D. Schonfeld, “Multilinear discriminant analysis for higher-order tensor data classification,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 12, pp. 2524–2537, 2014.

[77] R. C. Hoover, K. Caudle, and K. Braman, “Multilinear discriminant analysis through tensor-tensor eigendecomposition,” in *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2018, pp. 578–584.

[78] C. Ozdemir, R. C. Hoover, K. Caudle, and K. Braman, “Kernelization of tensor discriminant analysis with application to image recognition,” in *2022 21st IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2022, pp. 183–189.

[79] H. Higashi, T. M. Rutkowski, T. Tanaka, and Y. Tanaka, “Multilinear discriminant analysis with subspace constraints for single-trial classification of event-related potentials,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 10, no. 7, pp. 1295–1305, 2016.

[80] M. Jamshidi Idaji, M. B. Shamsollahi, and S. Hajipour Sardouie, “Higher order spectral regression discriminant analysis (hosrda): A tensor feature reduction method for erp detection,” *Pattern Recognition*, vol. 70, pp. 152–162, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0031320317301875

[81] H. Zhou, L. Li, and H. Zhu, “Tensor regression with applications in neuroimaging data analysis,” *Journal of the American Statistical Association*, vol. 108, no. 502, pp. 540–552, 2013, pMID: 24791032. [Online]. Available: https://doi.org/10.1080/01621459.2013.776499

[82] E. F. Lock, “Tensor-on-tensor regression,” *Journal of Computational and Graphical Statistics*, vol. 27, no. 3, pp. 638–647, 2018, pMID: 30337798. [Online]. Available: https://doi.org/10.1080/10618600.2017.1401544

[83] X. Li, D. Xu, H. Zhou, and L. Li, “Tucker tensor regression and neuroimaging analysis,” *Statistics in Biosciences*, vol. 10, no. 3, pp. 520–545, Dec 2018. [Online]. Available: https://doi.org/10.1007/s12561-018-9215-6

[84] Y. Pan, Q. Mai, and X. Zhang, “Covariate-adjusted tensor classification in high dimensions,” *Journal of the American Statistical Association*, vol. 114, no. 527, pp. 1305–1319, Jul 2019. [Online]. Available: https://doi.org/10.1080/01621459.2018.1497500

[85] B. Guo, L. E. Eberly, P.-G. Henry, C. Lenglet, and E. F. Lock, “Multiway sparse distance weighted discrimination,” *Journal of Computational and Graphical Statistics*, vol. 32, no. 2, pp. 730–743, Apr 2023. [Online]. Available: https://doi.org/10.1080/10618600.2022.2099404

[86] M. K. Kalaiah, A. Jude, and V. P. Malayil, “Effect of inter-stimulus interval on the acoustic change complex elicited with tone-complex and speech stimuli,” *Indian Journal of Otology*, vol. 23, no. 2, pp. 83–88, 2017.

[87] M. Bousse, O. Debals, and L. De Lathauwer, “A tensor-based method for large-scale blind source separation using segmentation,” *IEEE Transactions on Signal Processing*, vol. 65, no. 2, pp. 346–358, 2016.

[88] J. Wen, X. Fang, J. Cui, L. Fei, K. Yan, Y. Chen, and Y. Xu, “Robust sparse linear discriminant analysis,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 2, pp. 390–403, 2019.

[89] S. P. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[90] P. McCullagh, *Generalized linear models*. Routledge, 2019.

[91] K. Lange, J. Chambers, and W. Eddy, *Numerical analysis for statisticians*. Springer, 2010, vol. 1.

[92] M. Grant and S. Boyd, “CVX: Matlab software for disciplined convex programming, version 2.1,” https://cvxr.com/cvx, Mar. 2014.

[93] ——, “Graph implementations for nonsmooth convex programs,” in *Recent Advances in Learning and Control*, ser. Lecture Notes in Control and Information Sciences, V. Blondel, S. Boyd, and

H. Kimura, Eds. Springer-Verlag Limited, 2008, pp. 95–110, http://stanford.edu/~boyd/graph_dcp.html.

[94] B. Jemel, C. Achenbach, B. W. Müller, B. Röpcke, and R. D. Oades, “Mismatch negativity results from bilateral asymmetric dipole sources in the frontal and temporal lobes,” *Brain topography*, vol. 15, pp. 13–27, 2002.

[95] K. C. De Sousa, D. W. Swanepoel, D. R. Moore, H. C. Myburgh, and C. Smits, “Improving sensitivity of the digits-in-noise test using antiphasic stimuli,” *Ear and Hearing*, vol. 41, no. 2, pp. 442–450, 2020.