Tensor decomposition for data mining from brain electrical responses

MEng Dissertation

Divyansh Manocha
CID: 01053537

In Partial Fulfillment of the Requirements for the
Degree of
Electronic and Information Engineering

Imperial College London

Project Supervisor:	**Professor Danilo Mandic**
Second Marker:	**Professor Athanassios Manikas**
Advisors:	Ilya Kisil

Date of submission: 19 June 2019

Abstract

Motivation: Electroencephalography (EEG) data is a notoriously noisy signal which nevertheless is a very important keyhole into the neural function, as it measures brain activity at a very high temporal resolution. The state of the art techniques for the analysis of EEG employ two dimensional matrix analysis, such as Independent Component Analysis (ICA), to decompose the signal for the localisation/removal of activities. This restricted representation of the data in blind source separation (BSS) methods often requires the enforcement of unreasonable assumptions and does not allow for simultaneous time frequency analysis. The removal of biologically generated noise such as blinks (artifacts) is an active research area, which this study addresses using the superior properties of tensor decompositions.

Summary: With the objective of artifact removal the thesis explores a variety of tensor methods, including variants of Canonical Polyadic Decomposition (CPD), Tucker decomposition and Coupled Matrix Tensor Factorisation (CMTF). A quantification method for the analysis of the performance of each method is developed with the aim of measuring the retrieval of desired neural activity, in the case of synthetic and real world datasets. A unified method for artifact removal in all decomposition methods is also developed. The software developed in this study allowed vast expansion of an existing open source library with methods and algorithms that are not currently available in other tensor toolbox libraries. Careful optimisation was performed where required to allow trouble-free big data analysis.

Results: ICA, being the most popular current method, was established as a benchmark for the analysis. It was shown that methods based on tensor decomposition were able to better localise and remove artifacts; although tensors did not always provide a better absolute performance. It is further shown that tensors are able to extract activities of certain frequencies such as line noise power, removing the need for notch filtering. Tensor decomposition methods are recommended as a framework of choice for BSS as they enable superior insights into the physical meaning, and allow a unified approach to cleaning EEG signals.

Introduction

ElectroEncephaloGram (EEG) capture voltage fluctuations in the brain measured with the use of electrodes, in a non-invasive manner. An electrode measures the electrical potential with respect to a reference electrode. It is common to have one reference electrode for all other scalp electrodes. The first measurement of human EEG was performed by Hans Berger, a German neurologist, in 1924 to study the functioning of the human brain [1]. He later concluded that the data principally consisted of EEG signals. Unusual behaviour offers insights into neurological problems, one very common and successful application of which has been in the detection of epilepsy.

Artifacts in EEG: defining the problem

The signals are a superposition of the activity of multiple neural sources and noise composing of biological or technical artifacts and any undesirable distortions. The contamination in EEG may be generated from other voluntary or involuntary activities of a patient, such as blinking, muscle movements or hear beats (ECG rhythms), which are categorised as biological artifacts. They will be deemed as a type of noise in this study. Other technical artifacts may include from noise generated from the equipment, line noise, high voltage slow waves (HVS) or phase drifts. Unlike in many areas of signal processing, the desired signal here usually has a much smaller amplitude than many of the artifacts that it is contaminated with- which poses an interesting and complicated problem. Certain activities such as the ECG rhythms or line noise exhibit well defined frequencies in the spectral domain, and this study will not focus on these. Ocular artifacts are explored in this thesis, as localising them requires an analysis of multiple domains - a well suited problem to Tensors.

Blind Source Separation for EEG: defining the class of solutions

Time-frequency analysis, of the time varying potential differences, has become an essential tool in detecting neurological disorders. While spectral analysis gives in-

formation on the most dominant frequency spectra, temporal analysis is used to extract information of the locations of abnormal activities. It should be noted that spectral analyses groups signals into rhythms (alpha, beta, theta and delta). This is usually followed by a spatial analysis to localise the brain regions with these rhythms. ERP analysis is simply a time domain analysis to determine when a certain event occurs. Frequency domain analysis will analyse the spectral components of the dataset, typically by Fourier analysis or Wavelet analysis that attempts to address when certain frequencies occur. This was required since neurones are constantly oscillating and an analysis of long term periods, making temporal analysis alone difficult.

It is usually the case that decomposition methods, such as PCA (Principal Component Analysis) or ICA (Independent Component Analysis), are applied to filter relevant information. However such techniques make the assumption that the components are orthogonal to each other or that the noise signals are statistically independent to obtain a unique decomposition.

They are usually used to localise a particular problem by a predominantly qualitative analysis, and therefore allowing focused research on the particular area of the brain. EEG signals are traditionally represented as a vector or matrix with a channel and time mode, in order to allow for processing and analysis, such as that of detecting seizures or removing artifacts. Dimensionality reduction, feature extraction and spectral analysis are usually techniques that accompany this objective. However the raw datasets are inherently tensors of order greater than two modes: extended by multiple subjects, trials and a frequency domain representation. The restriction to two models requires unreasonable assumptions such as independence or orthogonality of sources, in order to achieve a unique decomposition. Tensor methods allow all three components, and perhaps more, to be considered simultaneously without the need of such strong conditions. In this study therefore EEG signals will be decomposed as tensors of more than two modes.

Tensor Decompositions: defining the proposed solution

Tensors, in the sense of multi-dimensional arrays, are multi-linear generalisations of matrices. Large scale, multi-model and heterogenous datasets are often intuitively represented as tensors. As a result, there have been numerous recent advancements regarding tensor representations for data mining. Matrix methods generally require reshaping or restricting the obtained data in order to perform two dimensional analysis, losing some inherent characteristics and therefore motivating the need for improvement.

Tensor decomposition has become increasingly important for data mining, as large

scale computational and storage resources become readily available. It is believed that their main advantage lies in keeping the inherent structure of the data intact to provide a better analysis of the multivariate dataset. This approach itself is not novel, and was first suggested by A. S. Field et al. with Parallel Factor (PARAFAC) decomposition [2].

The data obtained from each electrode, usually as a group of cells, are referred to as atoms in literature. Artifacts in the context of EEG datasets are activities, commonly involuntary, that generate signals in the brain - and therefore in the case of signal processing are errors. Tensor decomposition, and the removal of components that correspond to artifacts during reconstruction are ways of localising activity. This study explore numerous decomposition methods and evaluates their performance against a real world EEG dataset, as well as adding any tensor method implemented to a tensor toolbox library.

Other methods

Analysis using event-related potentials (ERPs) uses the high temporal resolution of the signal, and involves the use of multiple trials to *'lock'* to a stimulus. The selected fixed length segments of EEG must have synchronised events, called epochs. Often the simplest case with multiple trials is to reject ones that possess unwanted behaviour. This does not require extensive analysis, but would also significantly reduce the dataset size. In the majority of the thesis it will be assumed that only one trial is available, in order to develop an efficient technique that does not rely on the assumed availability of large datasets.

Acknowledgements

Throughout this project I have received a great deal of support. I would first like to thank my principal supervisor, Professor Danilo Mandic at Imperial College London, whose experience in the field of study was invaluable in directing my research. I am grateful that he was always happy to have technical discussions when I required assistance.

I would also like to thank my second marker, Professor Athanassios Manikas, whose expert opinions helped me to better formulate the research topic.

I am grateful to Ilya Kisil whose previous work, which the software engineering tasks of this project were built on, and up to date knowledge on the field were vital to the project.

In addition, I am also thankful to my parents and brother for their continued non-technical support and counsel.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 — Overview and Contributions

The primary aim of the project is to produce an analysis of sophisticated tensor methods, principally, for data mining with EEG dataset. It is anticipated that the research will allow for better analysis of brain activity, such as in localising areas of activity by removing artifacts through decomposition. The tensor manipulation algorithms that will be explored essentially divide themselves into the categories: higher order tensor decomposition methods, feature extraction and data fusion. All algorithms are to be implemented and the results to be shown using either a synthetic or real-world dataset, as an exploration of their properties and an evaluation of their uses. Thereby visualisation and testing will also be a key aspect in the continuous integration of this project.

The thesis is structured to introduce the tensor notation and decomposition methods that the techniques developed in this project build upon. The Tucker Decomposition (also known as HOSVD) is a direct generalization of PCA: it decomposes a tensor into a core tensor (the entries of which show the level of interaction between the different components) multiplied by a matrix along each mode. CP decomposes a tensor into a sum of rank-one tensors (combinations of which are factor matrices). The most common model for data fusion is the Coupled Matrix-Tensor Factorization (CMTF), which employs an existing tensor decomposition method such as Tuckers or CP. Chapter 3 provides a theoretical introduction to the research with a review of current literature. Chapter 4 then employs the properties of these algorithms to evaluate the most useful tensor methods in the context of EEG specifically. Using this analysis a hypothesis and investigation is built that describes why it is expected that certain tensor decomposition methods will produce favourable results to ICA. A further decomposition method, Parafac2, is introduced in this chapter with the hypothesis that it will outperform other tensor methods that have been previously employed in literature.

In Chapter 5 a method to quantify the performance of the algorithms was made, introducing metrics most useful in the case of EEG signals. A simpler approach with multiple performance measures was deemed better than a unified metric that

may lose information. Distinction between measuring the performance of datasets with known ground truth and unknown ground truth was made here. Large efforts were also placed in developing an ideal synthetic dataset, to enable a robust comparison of the tensor methods. The real datasets used were also briefly introduced.

Using ICA as a baseline metric, the results on the synthetic dataset are reported and discussed in Chapter 6. Results on the real datasets are further reported and evaluated in Chapter 7. The findings are summarised and an analysis of the decomposition methods is produced in Chapter 9.

There is an existing library that is being built in Imperial College London for tensor methods: HOTTBOX [3]. Any significant additions to the open source software developed as part of this project are briefly described in Chapter 8, with the hope that it will help further scientific contributions in the field of tensors.

2 — Definitions and Notations

An in-depth knowledge of basic linear algebra is assumed in this study, upon which some definitions in multi-linear algebra for tensors are defined. Some of the necessary definitions and notations which may not be familiar are defined below. These are assumed for the rest of this study.

A tensor is a multi-linear generalisation of a matrix or a vector, and is denoted by χ in this report. The dimensions (which also be referred to as the n^{th} order) are given by the number of modes (n^{th} mode is analogous to the term $n^{th} dimension$) of a tensor: $\chi \in \mathbb{R}^{I_1 \times I_2}$. Therefore a second order tensors is a matrix, whilst a first order tensor is a vector. It can also be said that this is the result of the tensor product of n vector spaces. Please note that while some relationship exists, this definition is different to the mathematical description of *tensor fields*.

2.1 Linear operations

The Kronecker product of two matrices $\boldsymbol{A} \in \mathbb{R}^{I \times J}$ and $\boldsymbol{B} \in \mathbb{R}^{K \times L}$ is a matrix $\boldsymbol{C} \in \mathbb{R}^{IK \times JL}$.

$$A \otimes B = \begin{bmatrix} a_{11}\boldsymbol{B} & a_{12}\boldsymbol{B} & \dots & a_{1J}\boldsymbol{B} \\ a_{21}\boldsymbol{B} & a_{22}\boldsymbol{B} & \dots & a_{2J}\boldsymbol{B} \\ \vdots & \vdots & \ddots & \vdots \\ a_{I1}\boldsymbol{B} & a_{I2}\boldsymbol{B} & \dots & a_{IJ}\boldsymbol{B} \end{bmatrix}$$

The entries of the matrix $\boldsymbol{C}$ can be written in the following way:

$$c_{(i-1)K+k;(j-1)L+l} = a_{ij} b_{kl} \tag{2.1}$$

The Hadamard product is obtained by multiplying the matrices $A \in \mathbb{R}^{I \times J}$ and $B \in \mathbb{R}^{I \times J}$ element-wise to give matrix $C \in \mathbb{R}^{I \times J}$

$$A * B = \begin{bmatrix} a_{11}b_{11} & a_{12}b_{12} & ... & a_{1J}b_{1J} \\ a_{21}b_{21} & a_{22}b_{22} & ... & a_{2J}b_{2J} \\ \vdots & \vdots & \ddots & \vdots \\ a_{I1}b_{I1} & a_{I2}b_{I2} & ... & a_{IJ}b_{I2} \end{bmatrix}$$

The Khatri-Rao product is similar to the Kronecker product, but it performs the product on the same columns. Therefore for $A \in \mathbb{R}^{I \times K}$ and $B \in \mathbb{R}^{J \times K}$ to give matrix $C \in \mathbb{R}^{IJ \times K}$

$$A \otimes B = \begin{bmatrix} a_{11}\boldsymbol{b_1} & a_{12}\boldsymbol{b_2} & ... & a_{1K}\boldsymbol{b_K} \\ a_{21}\boldsymbol{b_1} & a_{22}\boldsymbol{b_2} & ... & a_{2K}\boldsymbol{b_K} \\ \vdots & \vdots & \ddots & \vdots \\ a_{I1}\boldsymbol{b_1} & a_{I2}\boldsymbol{b_2} & ... & a_{IK}\boldsymbol{b_K} \end{bmatrix}$$

$$A \odot B = [\boldsymbol{a_1} \otimes \boldsymbol{b_1} \boldsymbol{a_2} \otimes \boldsymbol{b_2} ... \boldsymbol{a_K} \otimes \boldsymbol{b_K}] \tag{2.2}$$

2.2 Tensor Notation

Multi-linear algebra	
$\boldsymbol{x} = [x_n]$, $\boldsymbol{X} = [x_{n_1,n_2}]$, $\boldsymbol{\chi} = [x_{n_1,...,n_p}]$	vector, matrix and tensor
$\boldsymbol{x}^*$	Complex Conjugate
$\Lambda \in \mathbb{R}^{R \times R \times ... \times R}$	Diagonal core tensor with nonzero entries λ_r on main diagonal
X^{-1}, $X^\top$, $X^\dagger$	Inverse, Transpose and Moore-Penrose pseudo inverse
$\circ$	Outer product
$\otimes$	Kronecker product
$\odot$	Khatri-Rao product
$\times_p$	Mode-n product
$\boldsymbol{X}_{(n)} \in \mathbb{R}^{I_n \times I_1 \cdots I_{n-1} I_{n+1} \cdots I_N}$	Mode-n matricization of tensor χ
$[[A, B, C]]$	Canonical Polyadic (Parafac) decomposition
$\boldsymbol{\chi}(:, i_2, i_3, \cdots, i_N)$	Mode-1 fiber of a tensor
$\boldsymbol{\chi}(:, :, i_3, \cdots, i_N)$	Slice of a tensor

Table 2.1: Multi-linear algebra notation used in this project

Please note that, similar to the matrix notation, a colon represents all entries of a mode of the tensor $\boldsymbol{\chi}$.

Information Theory	
$x \perp y$	x orthogonal to y
$x \sim P$	x follow the distribution P

Table 2.2: Multi-linear algebra notation used in this project

The Frobenius Norm of a Tensor $\chi \in \mathbb{R}^{I_1 \times I_2 \ldots \times I_N}$ is given as follows:

$$\|\chi\| = \sqrt{\sum_{i_1}^{I_1} \sum_{i_2}^{I_2} \ldots \sum_{i_N}^{I_N} \chi(i_1, i_2, \ldots, i_n)^2} \tag{2.3}$$

Matricisation

N-mode Matricisation is simply the unfolding of a tensor to a matrix in N different ways. The N-mode product between a tensor $\chi \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ and a matrix is denoted as follows:

$$y(i_1, \ldots, i_{n-1}, r, i_{n+1}, \ldots, i_n) = \sum_{j=1}^{I_n} \chi(i_1, \ldots, i_{n-1}, r, i_{n+1}, \ldots, i_n) \boldsymbol{A}(j, r) \tag{2.4}$$

N-mode tensors and their products

Fixing indices of a particular tensor gives a subtensor, some of which have interesting properties. In the table 2.1 the definitions of a slice and fiber were introduced. Considering a third order tensor these can be visualised as shown in figure 2.1. In literature $\chi_{::k}$, $\chi_{i::}$, $\chi_{:j:}$ are referred to as front, horizontal and lateral slices respectively.

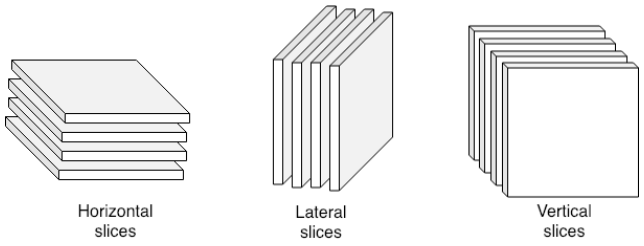

Figure 2.1: Slices of a tensor of order 3

The definition of a fiber is that all indices but one of a tensor are fixed. Therefore for a matrix, a column is defined to be a fiber. For tensors of order three, $\chi_{:jk}$, $\chi_{i:k}$, $\chi_{ij:}$ are mode-1, mode-2 and mode-3 fibers.

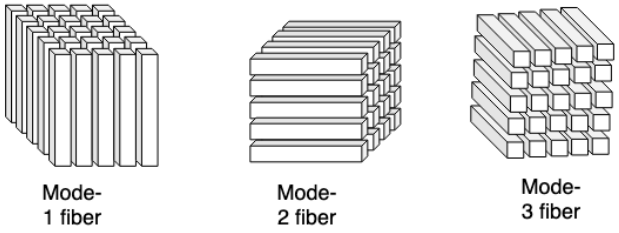

Figure 2.2: Fibers of a tensor of order 3

The mode-n product is simply the unfolding of a tensor by a mode followed by a matrix or vector multiplication. This is then reshaped into a tensor. Let $\boldsymbol{U}$ be a matrix $\in \mathbb{R}^{J \times I_N}$

$$(\chi \times_n \boldsymbol{U})_{i_1 \ldots i_{n-1} j i_{n+1} \ldots i_N} = \sum_{i_n=1}^{I_n} x_{i_1 i_2 \ldots i_N} u_{j i_n} \tag{2.5}$$

Unsupervised decomposition

Forward model: Matrices

$$\begin{aligned} \boldsymbol{x_t} &= \sum_{k=1}^{K} \boldsymbol{b}_k^T \boldsymbol{S}_{k,t} + \boldsymbol{e}_t \\ \boldsymbol{X} &= \boldsymbol{A}\boldsymbol{S} + \boldsymbol{E} \end{aligned} \tag{2.6}$$

Forward model: Tensors

$$\chi = \kappa \times_{i=1}^{N} \boldsymbol{U}^{(i)} + \boldsymbol{\epsilon} \tag{2.7}$$

3 — Background: Tensor Decomposition, Feature Extraction and Data mining

This chapter aims to provide an overview of related literature on low rank tensor decomposition, feature extraction, data fusion, and a brief analysis on the current developments for data mining.

3.1 Introduction

There exist very well studied and applied techniques of decompositions and feature extraction when concerning linear algebra. However, generalisations of such techniques to multi-linear algebra has only seen developments more recently. This survey aims to introduce these, as well as to determine which applications will be most appropriate for such datasets.

Some of the vast research areas where tensor decomposition has been used successfully include Chemometrics [4] [5], psychometrics [6] [7] [8], signal processing [9], computer vision [10], data mining and machine learning [11]. The generalised techniques have also seen more theoretical application in the development of multi-dimensional integrals, and multi-dimensional convolution, solving stochastic and parametric PDEs.

Consider a tensor χ of order d. The size of χ will clearly increase exponentially relative to the size of d. This is a major problem with large datasets in numerous applications, including but not limited to machine learning algorithms, and the problems that arise with this are often referred to as *the curse of dimensionality*. Therefore this motivates the first application that will be explored in this study: low-rank tensor decompositions of higher order tensors for a compressed representation. Whilst there are reshaping techniques that allow one to perform such analysis on these datasets in two dimensions, some inherent characteristics

are inevitably lost in a majority of the cases with such methods.

A concise assessment of the tensor decomposition methods has been presented in [12], and a more detailed analysis is found in [13] [14]. Literature reviews of a similar nature can be found in [15] [16], however the focus of this review is solely on data mining - that will be applicable to an EEG dataset for example.

3.2 Structured Tensors: tensorization by folding

Also known as segmentation, a tensor may be obtained through the folding of a vector by rearranging and reshaping data entries. In folding, a tensor χ is obtained from a vector $\boldsymbol{x}$ such that:

$$\begin{gathered} \chi(i_1, i_2, ..., i_N) = \boldsymbol{x}(i) \\ \forall 1 \leq i_n \leq I_n \\ where\ i = 1 + \sum_{n=1}^{N} (i_n - 1) \prod_{k=1}^{n-1} I_k\ is\ a\ linear\ index\ of\ (i_1, i_2, ...) \end{gathered} \tag{3.1}$$

Hankel Folding

Prior to discussing a Hankel tensor, introduced by Papy *et. al* [17] in a signal processing application, consider the structure of a Hankel matrix. It is a square matrix with the same ascending skew-diagonals.

An $I \times J$ matrix of length $I + J - 1$ is as follows:

$$\boldsymbol{A}_{I,J}(\boldsymbol{y}) = \begin{bmatrix} a_1 & a_2 & ... & a_J \\ a_2 & a_3 & ... & a_{J+1} \\ \vdots & \vdots & \ddots & \vdots \\ a_I & a_{I+1} & ... & a_L \end{bmatrix}$$

A Hankel tensor of order N, and therefore of size $I_1 \times I_2 \times ... \times I_N$, is obtained from a vector of length $\sum_n I_n - N + 1$ such that:

$$\chi(i_1, i_2, ..., i_N) = a(i_1 + i_2 + ... + i_N - N + 1) \tag{3.2}$$

It should be noted that any slice of the Hankel tensor (i.e. fixing $(N-2)$ indices) are Hankel matrices. If it is the case that $I_n = I\ \forall n$ with identical dimensions, then it is also a symmetric tensor. A Hankel tensor can be constructed from another Hankel tensor of a smaller order by converting its fibers to Hankel matrices.

Toeplitz Folding

A Toeplitz matrix has the same entries in each diagonal, such that it obtains the following structure from a vector $\boldsymbol{a}$ of length $I + J - 1$:

$$\boldsymbol{A}_{I,J}(\boldsymbol{y}) = \begin{bmatrix} a_I & a_{I+1} & ... & a_L \\ a_{I-1} & a_I & ... & a_{L-1} \\ \vdots & \vdots & \ddots & \vdots \\ a_1 & a_2 & ... & a_{L-I+1} \end{bmatrix}$$

The Toeplitz tensor can be derived by considering the discrete convolution between vectors, as stated in [14]. It is defined as follows:

$$\begin{aligned} \boldsymbol{\chi}(i_1, i_2, ..., i_N) = a(\bar{i_1} + \bar{i_2} + ... + \bar{i_{N-1}} + i_N) \\ where \bar{i_n} = I_n - i_n \end{aligned} \tag{3.3}$$

Convolution Tensor

Toeplitz and Hankel tensors enlarge the number of data entries of the original samples, and are therefore unsuitable for analysing signals of large sizes. Consider a third order tensor $I \times J \times K$ where $J = K - I + 1$, for which the $(l - I)^{th}$ diagonal elements of the l^{th} slice are all ones.

$$\boldsymbol{\chi}(:,:,l) = \begin{bmatrix} 0 & 0 & 0 & ... & 0 \\ & & & & \\ 1 & & \ddots & & \\ & \ddots & & & \\ 0 & ... & 1 & ... & 0 \end{bmatrix}$$

A further generalisation and properties are discussed in [18]

3.3 Low rank tensor decomposition

Whilst for second order tensors widely known and studied techniques exist (built on singular value decomposition), a generalisation to tensors is not trivial. The study will principally focus on three types of tensor decomposition: Canonical Polyadic (Parafac), Tucker and Tensor Train. A comprehensive overview of the methods will be provided, as well as their applications in data minings. The obtained low rank representations are particularly useful in separating signals from a mixture.

Canonical Polyadic (Parafac) Decomposition

The simplest decomposition of a tensor of Nth-order χ is into a linear combination of rank one components. [19]

$$\chi_{i_1,i_2,..,i_d} = u_1^{(d)} \otimes u_1^{(d-1)} \otimes ... \otimes u_1^{(1)} + ... + u_R^{(d)} \otimes u_R^{(d-1)} \otimes ... \otimes u_R^{(1)} \tag{3.4}$$

This is often expressed as $\chi \approx \sum_{r=1}^{R} \lambda_r b_r^{(1)} \circ b_r^{(2)} \circ ... \circ b_r^{(d)} = [[\Lambda; \boldsymbol{B^{(1)}}, \boldsymbol{B^{(2)}}, \boldsymbol{B^{(3)}}, ...]]$ where λ_r are entries of the diagonal core tensor $\Lambda \in \mathbb{R}^{R \times R \times R...}$ and $B^{(n)} = [b_1^{(n)}, b_2^{(n)}, b_R^{(n)}]$ are the factor matrices.

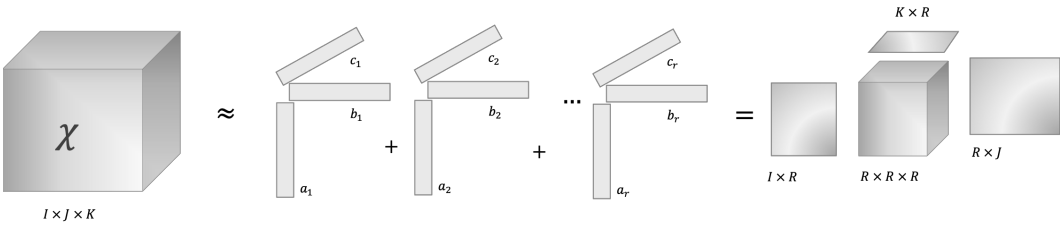

Figure 3.1: Visual representation of the Canonical Parafac decomposition

The rank of such a tensor is therefore defined to be the smallest number of rank one components that generate χ as their sum. If the elements of a tensor were to be drawn from a continuous uniform distribution, the typical rank is introduced as any rank that occurs with a non-zero probability. The maximum rank is introduced as the largest rank attainable. Finding the rank is an NP-hard problem, and no robust algorithms or estimations exists for tensors greater than third order. However rank decompositions are often unique for higher order tensors, which means CP decomposition is unique.

The notion of border rank is introduced to be the minimum number of rank one tensors that approximate the tensor with an arbitrarily small error.

CP Decomposition is estimated by minimising an appropriate cost function, which provides a representation is arbitrarily close in terms of fit. The cost function commonly used is that of the least squares type.

$$J_1(\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}...\boldsymbol{x}^{(N)}) = \|\chi - [[\boldsymbol{\Lambda}; \boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}...\boldsymbol{x}^{(N)}]]\|_F^2 \tag{3.5}$$

A commonly used method to compute this for a tensor is using the alternating least square algorithm, which individually optimises each component by setting the others to be fixed.

The component matrices are updated as follows:

$$\boldsymbol{A}^{(n)} \leftarrow \boldsymbol{X}_{(n)} \left(\odot \boldsymbol{A}^{(k)}\right) \left(\otimes (A^{(k)T} \boldsymbol{A}^{(k)})^{\dagger}\right) \tag{3.6}$$

Algorithm 1: CPD using ALS

Data: Tensor $\boldsymbol{\chi} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$, rank R
Result: $A^{(1)} \in \mathbb{R}^{I_1 \times R}, A^{(2)} \in \mathbb{R}^{I_2 \times R}, \ldots, A^{(N)} \in \mathbb{R}^{N \times R}, \boldsymbol{\lambda} \in \mathbb{R}^{1 \times R}$

Initialise factor matrices $\boldsymbol{A^{(n)}} \in$ **repeat**
  for $n \leftarrow k$ ***to*** N **do**
    $\boldsymbol{V} \leftarrow \boldsymbol{A}^{(1)T} \cdot \boldsymbol{A}^{(1)} * \cdots * \boldsymbol{A}^{(n-1)T} \cdot \boldsymbol{A}^{(n-1)} * \boldsymbol{A}^{(n+1)T} \cdot \boldsymbol{A}^{(n+1)T} * \cdots * \boldsymbol{A}^{(N)T} \cdot \boldsymbol{A}^{(N)}$
    $\boldsymbol{A^{(n)}} \leftarrow \boldsymbol{X}^{(n)} \left(\boldsymbol{A}^{(N)} \odot \cdots \odot \boldsymbol{A}^{(n+1)} \odot \boldsymbol{A}^{(n-1)} \odot \cdots \odot \boldsymbol{A}^{(2)} \odot \boldsymbol{A}^{(1)} \right) \boldsymbol{V}^{\dagger}$
    $\boldsymbol{\lambda} \leftarrow$ Norm of $\boldsymbol{A}^{(n)}$ columns
    Normalise $\boldsymbol{A}^{(n)}$ columns
  end
until *convergence criteria met*;
return $\boldsymbol{\lambda}, \boldsymbol{A}^{(N)}, \boldsymbol{A}^{(N-1)}, \ldots, \boldsymbol{A}^{(1)}$

The condition for uniqueness is that the the sum of rank of the factored matrices must be greater than a certain value. For example, for PARAFAC on a 3 way tensor, rank($\boldsymbol{A}$) + rank($\boldsymbol{B}$)+ rank($\boldsymbol{C}$) $\geq 2R + 2$ [20]. R is the number of components.

The time complexities for matrix unfolding and the Khatri-rao product are as shown below:

3-way case	$n \times n^2$	$r \times n^2$
d-way case	$n \times n^{d-1}$	$r \times n^{d-1}$

Some optimisations such as applying fast Johnson-Lindenstrauss Transform exist to ensure the incoherence. Implementation of a parallel ALS has also be suggested. [21]

Parafac can also be seen as a tucker model, introduced in section 3.3, in which the operation $\boldsymbol{\kappa} \times_{i=1}^{N} \boldsymbol{U}^{(i)}$ is restricted.

Applications

Two-way PCA in the context of multi-way arrays is more complicated than Tucker, which in turn is a more complex model than PARAFAC. Occam's razor is a very old principle, stating that the simplest model be preferred. The tensor method has therefore seen a very wide range of applications since its development.

Applications that require traditional PCA can generally be used with PARAFAC decomposition. It has been used extensively in chemometrics [4] [5], neuroscience [22] [23], data mining [11] [24], data fusion [25].

Tucker Decomposition

Tucker Decomposition may be viewed as a higher order principal component analysis. It decomposes a tensor into a core tensor multiplied by a matrix along each

mode, known as factor matrices. These are the principal components in each mode. Thereby providing a more general factorisation of the tensor.

$$\begin{aligned} vec(\boldsymbol{\chi}) &= (U_N \otimes U_{N-1} \otimes ... \otimes U_1) vec(\boldsymbol{\kappa}) \\ \boldsymbol{\chi} &\approx \sum_{r_1=1}^{R_1} ... \sum_{r_N=1}^{R_N} \kappa_{r_1, r_2 ... r_N} (\boldsymbol{u}_{\boldsymbol{r_1}}^{(\mathbf{1})} \circ \boldsymbol{u}_{\boldsymbol{r_2}}^{(\mathbf{2})} \circ ... \circ \boldsymbol{u}_{\boldsymbol{r_N}}^{(\boldsymbol{N})}) \\ &= \boldsymbol{\kappa} \times_1 \boldsymbol{U}^{(1)} \times_2 \boldsymbol{U}^{(2)} ... \times_N \boldsymbol{U}^{(N)} \\ &= [[\boldsymbol{\kappa}; \boldsymbol{U}^{(\mathbf{1})}, \boldsymbol{U}^{(\mathbf{2})}, ..., \boldsymbol{U}^{(\boldsymbol{N})}]] \end{aligned} \tag{3.7}$$

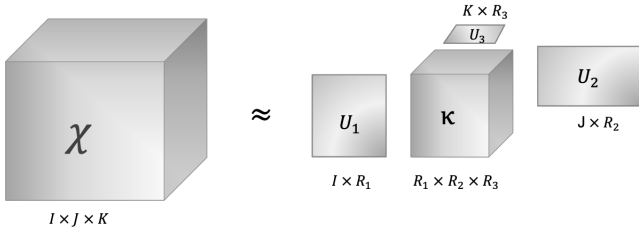

Figure 3.2: Visual representation of the Tucker decomposition

The projection orthogonal matrices $\boldsymbol{P}$, $\boldsymbol{Q}$, $\boldsymbol{R}$ are obtained by choosing a relative error ϵ, such that:

$$\|\boldsymbol{\chi} - (\kappa \times_1 \boldsymbol{P} \times_2 \boldsymbol{Q} \times_3 \boldsymbol{R})\| \leq \epsilon \|\chi\| \tag{3.8}$$

Tucker decomposition can be considered to be a more general form of the CP decomposition, where the core tensor is superdiagonal and $P = Q = R$. The $Tucker_{(n)}$ sets any subset of the factor matrices to the identity matrix. The core tensor can be viewed as a compression of the original tensor. Therefore any dataset that can be modelled with PARAFAC can also be modelled by Tucker. Unlike CPD, Tucker cannot be considered to be unique.

If all the factor matrices are full column rank, the decomposition is in an *independent format*. If they are orthogonal as well, the decomposition is of *orthonormal format*. Two of the most common algorithms for computing Tucker decomposition are the HOSVD model (Higher order singular value decomposition) or HOOI (Higher Order Orthogonal Iterations). Most implementations are based on ALS or HALS (hierarchical ALS).

The definition of *multilinear rank* of a tensor $\boldsymbol{\chi}$ is given as $(r_1, .., r_d)$. Unlike CP decomposition, at most r_μ of the set $T(r_1, ..., r_d)$ in a μ-rank tensor.

Consider the computation of Tucker decomposition. The least squares solution of the 3.7. HOSVD does not guarantee to reach the optimal solution to the decompo-

sition.

$$vec(\hat{\kappa}) = (U^{(N)} \otimes U^{N-1} \otimes ... \otimes U^{(1)})^{\dagger} vec(\chi) \tag{3.9}$$

By orthogonality $U^{(i)\dagger} = U^{(n)T}$, and we obtain the solution

$$\hat{\kappa} = \chi \times_N U^{(N)T} ... \times_2 U^{(2)T} \times_1 U^{(1)T} \tag{3.10}$$

On higher level perspective HOSVD calculates the singular matrices $\boldsymbol{U}^{(1)}, \boldsymbol{U}^{(2)}, ..., \boldsymbol{U}^{(N)}$, for every n-mode matricization of the tensor. Thereby giving an approximate solution to the problem

$$min_{\boldsymbol{B}} \|\chi - \boldsymbol{B}\|_F^2 \tag{3.11}$$

A study of the properties of HOSVD can be found in [26]. It was proven by De Lathauwer et al that every tensors admits a higher order singular value decomposition.

Numerous improvements to the efficiency of the algorithm exist. One common technique to obtain a cheaper approximation to the problem is the the Truncated HOSVD. [27] The orthogonal factor matrices are obtained from the SVD of the mode-k unfolding of the tensor, as has been shown above.

$$\chi^{(k)} = \boldsymbol{U}^{(k)} \Sigma^{(k)} \boldsymbol{V}^{(k)^T} \tag{3.12}$$

The fundamental concept behind T-HOSVD is to take the rank - $r_1, r_2, ..., r_d$and store only the top s_k left singular vectors $\boldsymbol{U}^{(k)}$ Computations of the the best mulit-linear rank based on the Newton-Grassmann method have also been explored by Lars Elden and Berkant Savas , which will not be discussed in this study. [26]

Algorithm 2: Higher-Order Singular Value Decomposition

Data: Tensor $\chi \in \mathbb{R}^{I_1 \times I_2 \times ... \times I_N}$, ranks $R_1, ..., R_N$
Result: $\boldsymbol{U}_1 \in \mathbb{R}^{I_1 \times R_1}, \boldsymbol{U}_2 \in \mathbb{R}^{I_2 \times R_2}, ..., \boldsymbol{U}_N \in \mathbb{R}^{I_N \times R_N}$ and $\mathcal{K} \in \mathbb{R}^{R_1 \times ... \times R_N}$

```
for n ← 1 to N do
    [U, σ, V] ← SVD(X_(n))
    U_n ← U(:, 1 : R_n)
end
K ← χ ×_N U_N^T ×_{N-1} ··· ×_1 U_1^T
return U_1, U_2, ..., U_N and K
```

Applications

The limitations of the algorithm primarily lie in the exponentially growing memory required to store the $r_1 \times ... \times r_d$ core tensor. For a massive scale dataset where the dimensions of d are large, this becomes too expensive.

Algorithm 3: Higher-Order Orthogonal Iteration

Data: Tensor $\boldsymbol{\chi} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$, ranks $R_1, \ldots, R_N$
Result: $\boldsymbol{U}_1 \in \mathbb{R}^{I_1 \times R_1}, \boldsymbol{U}_2 \in \mathbb{R}^{I_2 \times R_2}, \ldots, \boldsymbol{U}_N \in \mathbb{R}^{I_N \times R_N}$ and $\boldsymbol{\mathcal{K}} \in \mathbb{R}^{R_1 \times \ldots \times R_N}$

```
Initialise U_1...U_N using HOSVD or randomly
repeat
    for n ← 1 to N do
        W ← χ ×_N U_N^T ··· ×_{n−1} U_{n−1}^T ×_{n+1} U_{n+1}^T ··· ×_1 U_1^T
        [U, Σ, V] ← SVD(W_(n))
        U_n ← U(:, 1 : R_n)
    end
until convergence criteria met;
K ← χ ×_N U_N^T s ×_{N−1} ··· ×_1 U_1^T
return U_1, U_2, ..., U_N and K
```

Tucker decomposition has seen numerous applications in data mining and machine learning [28] [24], image processing [29], pattern recognition [30], signal processing [31] [9]. It is a fundamental decomposition technique for other methods, such as hierarchical Tucker decomposition and Tensor Train decomposition. As with many decomposition methods, it can be used for dimensionality reduction in order to compute CPD with better efficiency.

Tensor Train Decomposition

Tensor train decomposition attempts to address the major limitation of Tucker's decomposition method. The tensor is decomposed to the product of a matrix, three-mode core tensors (also referred to as 'transfer'), and another matrix. Every transfer core tensor is linked to its neighbour by a reduced mode R_i, which need not be calculated. The algorithm is stopped when the approximation error reaches a certain threshold instead.

$$\boldsymbol{\chi}(i_1, \ldots, i_d) \approx \sum_{r_1, r_2, \ldots, r_{d-1}} \boldsymbol{B_1(i, r_1)} \boldsymbol{\kappa}_2(r_1, j, r_2) \ldots \boldsymbol{\kappa}_{d-1}(r_{d-2}, d-1, r_{d-1}) \boldsymbol{B_d(r_{d-1}, d)} \quad (3.13)$$

It can also be represented as:

$$x_{i_1, i_2, \ldots, i_N} \approx \boldsymbol{G}^{(1)}_{\boldsymbol{i_1}} \boldsymbol{G}^{(2)}_{\boldsymbol{i_2}} \ldots \boldsymbol{G}^{(N)}_{\boldsymbol{i_N}}$$
$$\textit{where} \quad (3.14)$$
$$\boldsymbol{G}^{(n)}_{\boldsymbol{i_n}} = \boldsymbol{\chi}^{(n)}(:, i_n, :) \in \mathbb{R}^{R_{n-1} \times R_n}$$

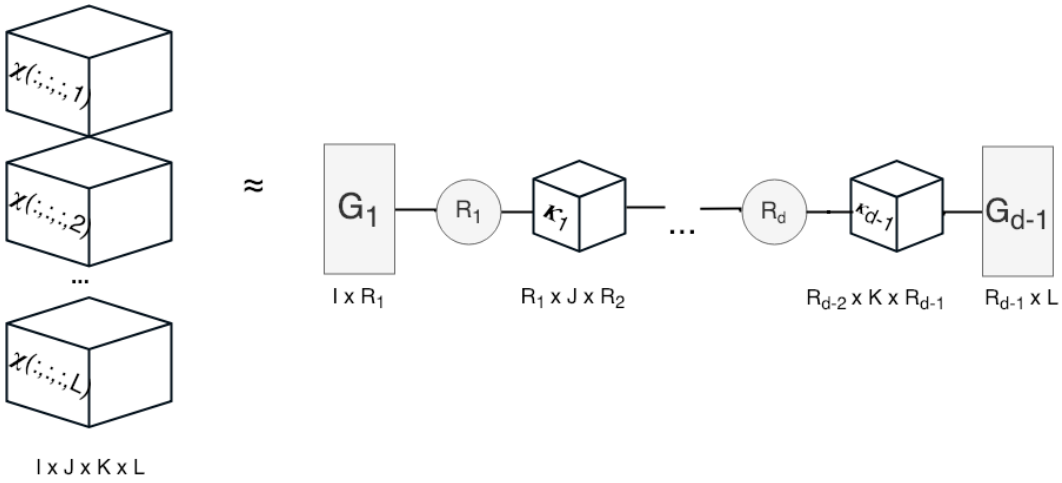

Figure 3.3: Visual representation of the Tensor Train decomposition

Now consider the notion of a rank in Tensor Train representation, which is determined by its matricization. Reshaping the data of the tensor χ into a $(n_1 ... n_\mu) \times (n_{\mu+1} ... n_d)$ will give a matrix of the dimensions $X^{(1,...,\mu)}$, implying from 3.14 that the rank of it is $\leq r_\mu$ where $\mu = (1, ..., d)$.

The simplest method of computing the tensor train decomposition is using the TT-SVD algorithm. As described in [32], the decomposition uses d sequential Singular Value Decompositions of auxiliary matrices. This becomes a recursive process to obtain the core tensors $\boldsymbol{G}$ as follows:

$$\chi(\alpha_1 i_2, i_3, ..., i_d) = \sum_{\alpha_2=1}^{r_2} \boldsymbol{G}_2(\alpha_1, i_2, \alpha_2) \chi'(\alpha_2 i_3, i_4, ..., i_d) \tag{3.15}$$

Applications

Tensor Train networks have wide applications that are not only limited to approximating tensorized vectors, but also matrices and scalar multivariate functions. They are also used in other fields of study, such as quantum physics where the representation are known as Matrix Product State [33]. The decomposition is commonly referred to as the density matrix renormalization group method [34].

Implementing mathematical operations, such as addition or the hadmard product, on tensors in the TT format is relatively simple. It has therefore seen successful applications in Markov Random Fields, used for image segmentation, where it is used for the partition function estimation and MAP inference. It was originally applied by Novikov *et. al* [35]. Another very useful application is in the field of machine learning. In the case of neural networks, most parameters are stored in the fully

connected layers and therefore similar accuracies to deep NNs can be achieved with a shallow network containing a large fully connected layer. Therefore Tensor Train are used to compress fully connected layers of neural networks. For details of the application refer to the paper [36].

3.4 Feature Extraction and Selection

Assume K data matrices $X^{(k)}$ that belong to C different classes. Feature extraction for the training samples involves reducing the model by applying simultaneous matrix factorisation.

$$\boldsymbol{X}^{(k)} \approx \boldsymbol{B}^{(1)}\boldsymbol{F}^{(k)}\boldsymbol{B}^{(2)T}, (k = 1, 2, ..., K) \tag{3.16}$$

Where $\boldsymbol{B}$ are the basis matrices that samples $\boldsymbol{F}^{(K)}$ represent the extracted features. If the feature matrix is a positive definite diagonal matrix and $B^{(i)}$ orthogonal, HOSVD is used. If the factors $A^{(i)}$ are orthogonal and feature matrices are dense, a DEDICOM (Decomposition into Directional Components) model is more appropriate.

The basis matrices are estimated by finding N common factors from d simultaneous decompositions of the d tensors $\boldsymbol{\chi}^{(k)} \in \mathbb{R}^{I_1 \times I_2 \times ... \times I_N}$.

$$\boldsymbol{\chi}^{(k)} \approx \boldsymbol{\kappa}^{(k)} \times_1 \boldsymbol{B}^{(1)} ... \times_N \boldsymbol{B}^{(N)}, (k = 1, 2, ..., d) \tag{3.17}$$

This is solved as follows:

$$arg_{\{\boldsymbol{B}^{(1)},...,\boldsymbol{B}^{(n)}\}} min \sum_{k=1}^{d} \|\boldsymbol{\chi}^{(k)} - \boldsymbol{\kappa}^{(k)} \times_1 \boldsymbol{B}^{(1)} ... \times_N \boldsymbol{B}^{(N)}\|_F^2 \tag{3.18}$$

Discriminant analysis To obtain the N orthogonal factors $\boldsymbol{U}^{(n)}$, and the core tensor κ, Tucker's HOOI algorithm was introduced by De Lathauwer et al. It is discussed in further detail in [37]. The training features obtained from the above can be used for discriminant analysis to project the training features onto the discriminant subspaces. A linear discriminant analysis approach will be used to find the discriminant projection matrices ψ for the feature set $\boldsymbol{g}$.

$$f^{(k)} = \boldsymbol{\psi} vec(\boldsymbol{\kappa}^{(k)}) \tag{3.19}$$

Where $f^{(k)}$ are the discriminant features, $\kappa^{(k)}$ the core tensor. The projection matrix ψ is found by maximising the fisher discriminant criterion:

$$\begin{aligned} \psi &= argmax_{\psi} \frac{\boldsymbol{\psi}^T \boldsymbol{S}_b \boldsymbol{\psi}}{\boldsymbol{\psi}^T \boldsymbol{S}_w \boldsymbol{\psi}} \\ &= argmax_{\psi} tr\left[\boldsymbol{\psi}^T (\boldsymbol{S}_b - \gamma \boldsymbol{S}_w) \boldsymbol{\psi}\right] \end{aligned} \tag{3.20}$$

$$\boldsymbol{f_{(k)}} = \boldsymbol{\psi}^T vec(\boldsymbol{\kappa}^{(k)}) \tag{3.21}$$

which is solved using the generalised eigenvalue decomposition (GEVD).

3.5 Data Fusion

The motivation behind data fusion is to develop a model that incorporates the information from a tensor χ along with *'extra'* information from other matrices or tensors. Thereby allowing multiple sources to be combined. The 'extra' information is usually meta data coupled with the main tensor. The most popular model, made by Acar et al. [38], is the Coupled Matrix-Tensor Factorization (CMTF). Assuming a coupled matrix and CPD has been implemented on the tensor to give matrices $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{D}$.

$$min_{\boldsymbol{a_r}, \boldsymbol{b_r}, \boldsymbol{c_r}, \boldsymbol{d_r}} \|\boldsymbol{\chi} - \sum_{r=1}^{R} \boldsymbol{a_r} \circ \boldsymbol{b_r} \circ \boldsymbol{c_r}\|_F^2 + \|\boldsymbol{Y} - \sum_{r=1}^{R} \boldsymbol{a_r} \boldsymbol{d_r}^T\|_F^2 \tag{3.22}$$

Solving this problem is very similar to that of traditional CP decomposition, by defining an Alternating Least Squares algorithm. The difference is that the data containing the meta information needs to be concatenated with the matricized tensor of χ: $\boldsymbol{X}^{(1)}$. As in for CPD, a gradient-based approach may also be employed.

In the case that certain datasets differ largely in terms of size to the others, they will predominantly be responsible for the approximation error. In such a case, weighting must be introduced.

One problem with this approach is that in the case of unshared factors, the decomposition does not represent the dataset well. Acar et. al developed a 'structure-revealing' model, which is also able to identify shared and unshared factors [39].

Note it is also possible to perform data fusion using Tucker decomposition instead of CPD.

An efficient version of the algorithm is found in [40], and the implementation used shown in listing 9.

Algorithm 4: Coupled matrix Tensor Factorisation: ALS

Data: Tensor $\chi \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$, Matrices $\boldsymbol{Y}^{(n)} \in \mathbb{R}^{I_n \times J_n}$, rank R
Result: Factors of tensor: $A^{(n)} \in \mathbb{R}^{I_n \times R}$, Factors of matrices: $\boldsymbol{B}^{(n)} \in \mathbb{R}^{J_n \times R}$

Initialise factor matrices $\boldsymbol{A^{(n)}} \in \mathbb{R}^{I_n \times R}, \boldsymbol{B^{(n)}} \in \mathbb{R}^{I_n \times R}$ **repeat**
  for $n \leftarrow k$ ***to*** N **do**
    $\boldsymbol{V} \leftarrow \boldsymbol{A}^{(1)T} \cdot \boldsymbol{A}^{(1)} * \cdots * \boldsymbol{A}^{(n-1)T} \cdot \boldsymbol{A}^{(n-1)} * \boldsymbol{A}^{(n-1)T} \cdot \boldsymbol{A}^{(n+1)T} * \cdots * \boldsymbol{A}^{(N)T} \cdot \boldsymbol{A}^{(N)} + \boldsymbol{B}^{(n)T} \cdot \boldsymbol{B}^{(n)}$
    $\boldsymbol{A^{(n)}} \leftarrow [\boldsymbol{X}^{(n)} : \boldsymbol{Y}^{(n)}] \left[\left(\boldsymbol{A}^{(N)} \odot \cdots \odot \boldsymbol{A}^{(n+1)} \odot \boldsymbol{A}^{(n-1)} \odot \cdots \odot \boldsymbol{A}^{(2)} \odot \boldsymbol{A}^{(1)} \right)^T : \boldsymbol{B}^{(n)T} \right]^T \boldsymbol{V}^{\dagger}$
    $\boldsymbol{\lambda} \leftarrow$ Norm of $\boldsymbol{A}^{(n)}$ columns
    Normalise $\boldsymbol{A}^{(n)}$ columns
  end
until *convergence criterion met*;
return $\boldsymbol{\lambda}, \boldsymbol{A}^{(N)}, \boldsymbol{A}^{(N-1)}, \ldots, \boldsymbol{A}^{(1)}$

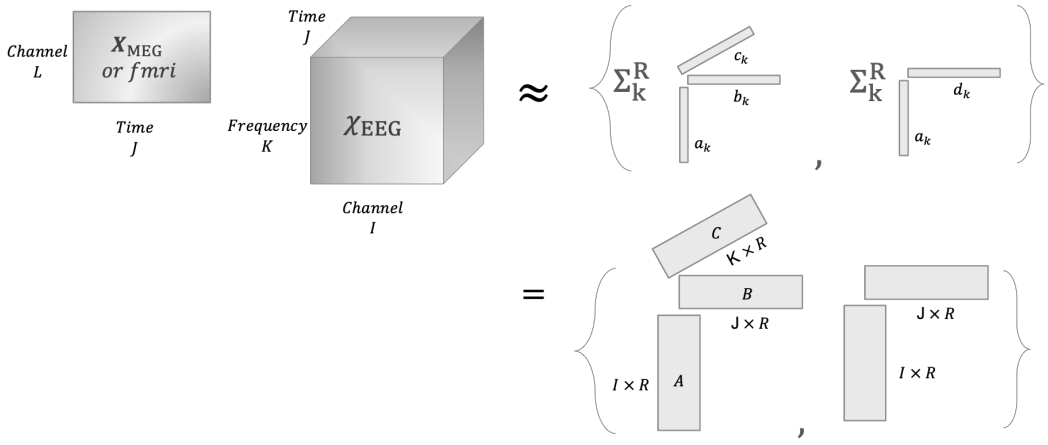

Figure 3.4: Visual representation of Coupled Matrix Tensor Factorisation using CPD.

Applications

Data fusion was introduced by Smilde *et al.* [41] in the context of chemometrics. It was later applied to data mining by Banerjee *et al.* [42]. Tensors and matrices have been coupled in previous research by Lin *et al.* [43] and Acar *et al.* [38].

3.6 Tensor methods on EEG datasets

The first study on event related potential was conducted by J. Mocks [2] in 1988, which was later proven to be equivalent to PARAFAC. There have been a small number of published work employing tensor methods on brain activity data. Early work conducted by Estienne *et al.* [23] studied decomposition of EEG data using Tucker, to extract information on the effects of a drug on brain activity. Miwakeichi *et al.* [22] produced a combined analysis of space-time-frequency, as will

be considered in this project. The tensor was composed after applying wavelet transformation to the data, and PARAFAC decomposition was used for analysis. More recently Acar *et al.* [44] conducted a similar study, also using PARAFAC and CWT (Continous Wavelet Transform), focused on localising epilepsy. All studies achieved promising results, allowing for a more quantitative and therefore automated approach to a dataset that is otherwise primarily qualitatively observed.

Consider now the representation of an EEG tensor. As it was originally introduced, a three mode tensor was formed consisting of time samples, scales and electrode channels. A depiction can be seen in figure 3.5. However the tensor methods are not in any manner restricted to these modes, and most information about the experiments can be incorporated. For example Cong *et al.* [45] give an example of the possibility of an EEG tensor with seven modes. This would consist of time, frequency, space, trials, subject, group and conditions.

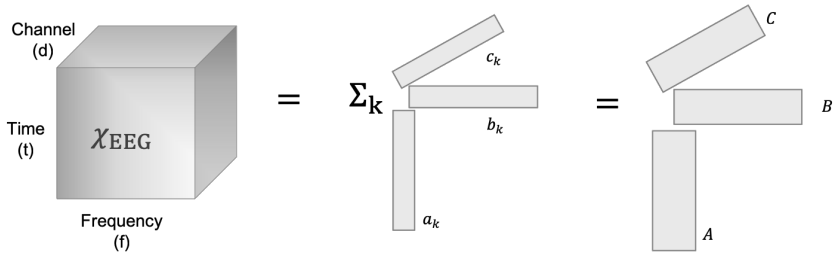

Figure 3.5: Decomposition of a multichannel EEG spectrum Tensor into a sum of 'atoms'.

For the purpose of obtaining a space/time/frequency analysis, either the Short Time Fast Fourier transform or a wavelet transform may be applied. The latter has been preferred in literatures due to its preservation of temporal properties. This is discussed in greater detail in Chapter 4.3. There seems not to have been an exploration of decomposition methods in previous literatures for EEG, which motivates this research.

4 — Theoretical analysis of tensor methods benefiting EEG properties

The method of extracting a certain activity (neural activity) from a signal (EEG), usually through the suppression of noise, is known as Blind Source Separation as these methods lack ground truth. Due to the difficult nature of the problem, very few literature studies have focused on quantifying the performance of Blind Source Separation methods. Though some analysis is inevitably conducted qualitatively, an attempt has been made in Chapter 5 to develop a method for quantifying the performance of artifact removal techniques. As EEG is a difficult dataset to analyse it is important to study the properties that can be exploited, and therefore determine whether tensor decomposition methods would have any advantage over the widely accepted Independent Component Analysis.

4.1 Statistical analysis of EEG and Artifacts

An Epoch in EEG is simply an extracted time window from the continuous signal, that is time locked relative to an event/baseline. Traditional means of analysis involve the exploration of Event Related Potentials (ERPs). ERP components are formed by averaging across epochs, in which the contributions of random activity cancel out as the number of trials are increased, leaving only systematic deflections relative to a baseline. This is an alternative approach to time frequency analysis, and will not be adopted in this project. ERPs do not allow the parallel cerebral activities to be processed.

Early advancements in the processing of such signals was using regression, where it was eventually found to have little success with certain types of artifacts, such as muscular. This was followed by principal component analysis, where Lins et. al removed blink artifacts [46].

The signals of particular interest are generated as action potentials at the membrane of a cell, where a process of depolarisation and repolarisation occurs with frequencies between 1 Hz and 100 Hz, with a range of amplitudes from $10\mu V$ to $100\mu V$ [47]. It is important to note, for context, that in order for a signal to be captured 10,000 to 100,000 neurons have to be activated simultaneously. Apart from blinks, the origin of artifacts may be from muscle activity (EMG) or the heart (ECG), which will be useful in Chapter 5. EEG itself is split into rythms in neuroscience literatures according to their frequencies: alpha, beta, theta and delta waves. The frequency range of alpha waves is between 8 Hz and 13 hz and is usually higher in amplitude. Beta rythms have a range between 13 Hz and 30 Hz; theta between 4 Hz and 8 Hz; delta below 3 Hz. Since each of these waves are predominantly formed from particular areas of the brain, it is known that the amplitude of signals across different areas of the scalp varies. The characteristics are also known to differ from one subject to another.

Considering their statistical properties, EEG signals are non-stationary, non-linear and not Gaussian. Numerous studies have shown that EEG signals are spatially correlated to only their neighbouring channels [48]. The small amplitudes of the *'true'* EEG signals make measured EEG very sensitive to artifacts.

4.2 ICA

Independent Component Analysis performs a full rank matrix factorization into statistically independent components. These components will be independent in the temporal domain, but spatially fixed. In its matrix representation this is denoted as shown in equation 4.1.

$$\begin{aligned} \boldsymbol{X} &= \boldsymbol{B} \cdot \boldsymbol{c} \\ &= \sum_{i}^{R} c_i b_i^T \\ &= \sum_{i}^{R} \boldsymbol{Y_i} \end{aligned} \tag{4.1}$$

where $\boldsymbol{X}$ is the combined signal $\in \mathbb{R}^{N_c \times N_t}$. $\boldsymbol{C} \in \mathbb{R}^{N_c \times R}$ contains the variable components (channel signature). $\boldsymbol{B} \in \mathbb{R}^{N_t \times R}$ contains the time signatures. N_c is the number of components, N_t is the number of time samples, R is the selected number of components in the decomposition. Note that $\boldsymbol{Y}_i$ are rank one components. The second line of equation 4.1 shows that the time signatures can only vary by a scalar, determined by the channel signature. The forward/reverse problem and the specifics of representing EEG are discussed further in Chapter 5.

ICA also assumes the PDFs are not Gaussian, along with statistical independence - though this is a weaker condition. To derive meaningful statistical properties (such as kurtosis) one also needs to make the assumption that the signal is statistically stationary. Finally only two modes can be used (e.g. time and space) which give components that are not necessarily unique. It is possible to resolve this, for example by allowing rotations of axis, but at the cost of further reinforcing the assumptions. Despite the assumptions ICA has been very successful as an artifact removal technique for noises that present themselves often and coherently, such as eye movements. Note the isolation of the components corresponding to artifacts is conducted through qualitative examination in this study.

There are a number of reasons why ICA performs better than a similar technique, also based on SVD, PCA. Firstly the decomposed projections sum linearly onto the electrodes, due to the independence property. This correlates to measured EEG, which is a linear superposition of the measured electrodes. Independence is not a realistic assumption, However it should be noted that artifacts such as muscle contractions are activated cognitively, and are therefore unlikely to be represented by the same components. This makes the assumption for EEG not entirely unreasonable. Finally it is highly unlikely that the sparsely active signals are Gaussian distributed.

The assumptions and the reasons why they might not be too strong are enlisted in table 4.1.

Assumptions	**Practical significance**
Independent time signals: $x_i \perp\!\!\!\perp x_j \forall i \neq j$	Signal components, including activity and noise, are not time locked to the sources of EEG activity.
Optimal performance when linear mixing	Volume conduction (even of the scalp) is considered linear.
No propagation delays	Signals are sent almost instantaneously, though instruments themselves can have time drifts.
One source per sensor. ICA cannot separate more than N sources for N sensors.	Data dependent. It may be the case that there are more than one statistically independent signals in a channel.

Table 4.1: Assumptions with Independent Component Analysis for uniqueness, and their plausibility/significance in EEG analysis.

From a geometric perspective, the columns of the weight matrix $\boldsymbol{C}^{-1}$ represent the

projection strength of each component to the electrode. The directions of two axis are selected such that the linearly transformed data has maximal entropy. This is no different to making the density of the data maximally uniform. An illustration is provided in Figure 4.1.

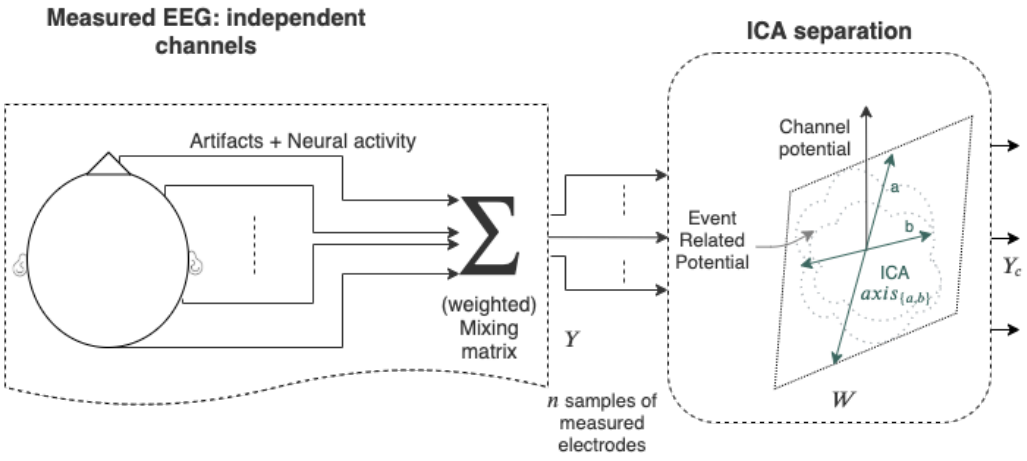

Figure 4.1: Illustration of activity removal using Independent Component Analysis in EEG signals.

4.3 Spectral Analysis

Non parametric approaches, such as the Welch method found in [49] which is an average of periodograms, require windowing and thus suffer from spectral leakage. This is a particular problem for EEG signals due to their small amplitudes. However parametric approach also assume the signal is statistically stationary which, as has been established, is not the case. In this project the two linear spectral methods that will be considered are Short Time Fourier Transforms (STFT) and Wavelet transforms.

STFT

Unlike in the case of Fourier transforms, STFT splits the signal ($x(t)$) into n number of segments and a window function $w(t)$ of the same width n is applied, as shown in equation 4.2. Therefore the assumption is now that each segment is stationary.

$$STFT^{(w)}(t,f) = \int_{-\infty}^{\infty} \left(x(t)w^*(t-t')\right) e^{-2\pi j f t} dt \tag{4.2}$$

An appropriate window function is one that would minimise spectral leakage for a particular signal. The assumption may be valid for small time windows, however it is unlikely to be the case for larger window sizes in EEG. STFT performs temporal

localisation of the Fourier spectrum $x(t)$ using the shifted window function of fixed duration and shape. The significance of this is that the frequency resolution and time resolution are fixed in the time frequency plane. Therefore there is also a trade off to be made between frequency resolution and time resolution, as a wide window size provides better spectral resolution and narrower windows lead to a better temporal resolution. This issue is addressed by wavelet transforms.

Wavelet transforms

A wavelet is a function of finite duration and energy, which is correlated to obtain the coefficients of the transform. A defined *mother wavelet*'s coefficients are evaluated at all time instants by translating across time samples, and is used as a reference. This is performed for every phase where the wavelet is scaled at another width and normalised to have the same energy as the mother wavelet. Therefore the coefficients are functions of position and scale.

Faster algorithms, such as discrete wavelet transforms exist. This may be used for implementation, but the algorithm itself does not provide any more insight into EEG datasets.

The wavelets are scaled to have long temporal durations and a large value for the scale parameter when the frequency range is low, addressing the issue with STFT mention above. Continous Wavelet Transforms (CWT) give the best representation for signals of high frequency, where a large proportion follows a gradual change with rare periods of short bursts. They will have limited resolution in low frequency regions of the EEG signal. An illustration of this can be found in 4.2.

Both methods can be used to extract alpha, beta, theta and gamma frequencies; however considering the properties discussed above, the rest of the project will primarily use wavelet transforms as the spectral analysis unless mentioned otherwise.

EEG signals are non-stationary, and therefore the frequency content may vary over time. Linear methods for Time Frequency Analysis (TFA) are based on a linear integral transformation of the signal, fundamentally quantifying similarity between the signal and a basis function. Unlike only spectral analysis, it is able to also provide insights into temporal evolution.

Quadratic methods

These methods use the signal $x(t)$ directly, instead of transforming it. They are built on the basis of the Temporal Correlation Function, shown in equation 4.3

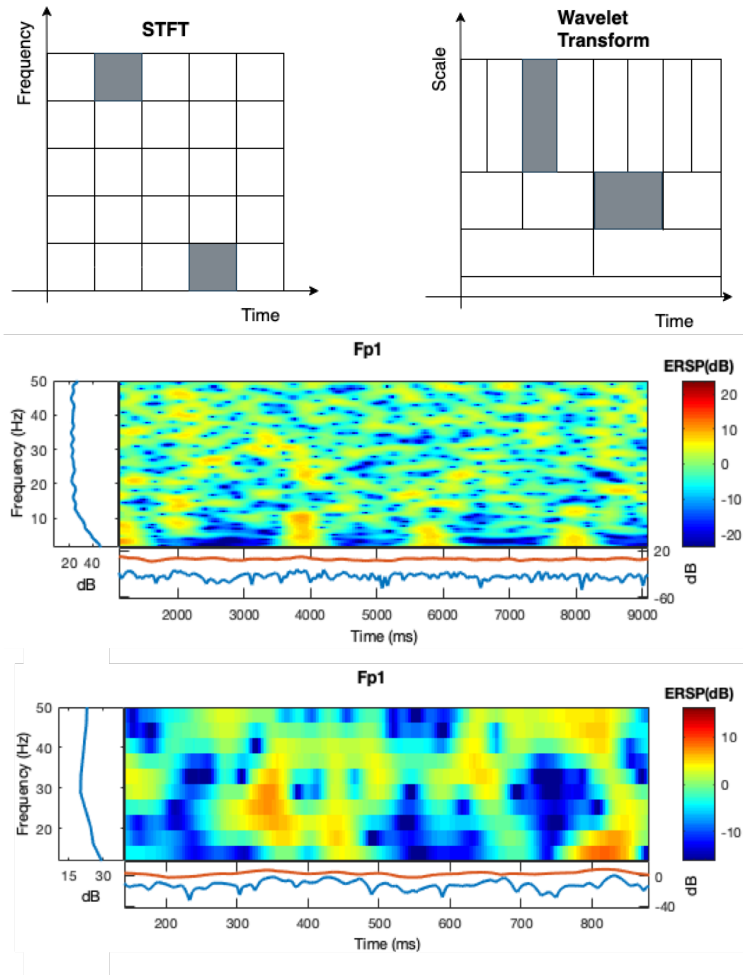

Figure 4.2: (a) Time and spectral resolution in STFT are fixed, whilst they can be altered in Wavelet Transforms. As can be seen for Wavelet Transforms, increasing the resolution of one decreases the resolution of the other. (b) Spectogram plot of FP1 channel from EEG with at two different time resolutions, demonstrating a clear difference in frequency resolution. The real (reading) dataset used in this project has been depicted with a morelet wavelet.

$$q_x(\tau, t) = x(t + \frac{\tau}{2}) \cdot x^*(t - \frac{\tau}{2}) \tag{4.3}$$

Wigner-Ville Distribution [50] is simply the Fourier Transform of equation 4.3. It was introduced by Ville [51] and Wigner [52] in 1948 and 1932 respectively. They are able to represent more information than spectrograms.

$$W_x(t, f) = \int_{-\infty}^{\infty} q_x(t, \tau) e^{-jw\pi f \tau} d\tau \tag{4.4}$$

The principal advantage of this family of distribution is the separation of frequency and time resolutions, at the expense of producing cross terms in $W_x(t, f)$. Cohen [53] introduced time frequency distributions (TFDs) based on WVD, which led to

the development of Reduced Interference Distribution (RID) as their kernelised version. For a complete description, the reader is referred to [54].

For the purpose of this project, wavelet transformations will be employed in analysis as they have been extensively studied in literature. Apart from their well-known properties, they are also simpler to implement than the alternative quadratic methods.

4.4 Family of Parafac Decomposition

CPD

One property of PARAFAC that is often exploited is that the decomposition is unique under weak conditions, as discussed in chapter 3. The atoms from the decomposition must be of rank one, which imposes a constraint on the temporal nature of the component signals. Consider the rewritten PARAFAC representation formed in Section 3.3, rewritten in equation 4.5.

$$\begin{aligned} \chi &= \sum_{i=1}^{R} \mathcal{Y}_i \\ &= \sum_{i=1}^{R} a_i \odot b_i \odot c_i \end{aligned} \tag{4.5}$$

For a particular channel p, $\mathcal{Y}$ can be written as shown in equation 4.6.

$$(\mathcal{Y}_i)_{channel\ p} = \boldsymbol{a_i} \cdot c_{i,p} \cdot \boldsymbol{b_i^T} \tag{4.6}$$

Note the notation $\chi_{channel p}$ refers to the p^{th} electrode in the channel mode

Here it is clear that $\boldsymbol{a_i} \cdot \boldsymbol{b_i^T}$ represents the time frequency distribution, as was the case with ICA. Equation 4.6 shows that for PARAFAC each channel will only vary by a scale factor: c_i. However, unlike ICA, non-stationarity to a certain extent can now be represented as long as the time frequency distribution is of rank one. This makes it more appropriate for EEG signals in theory.

Assumptions: For PARAFAC, let $A = [a_1,...,a_R], B = [b_1,...,b_R] and C = [c_1,...c_R]$. From equation 4.5 it is not difficult to see that the rank of these three matrices must be greater than one. This is a very weak assumption that is highly unlikely to be violated by EEG datasets, because no two electrodes would measure the same signals for their channels.

Imperfect reconstructions: There are a few reasons PARAFAC may not be able to represent a signal tensor truly. The afore mentioned notion of stationarity is

required for the observed signals which, to some degree, can be overcome using small windows. This is similar to the discussion for ICA. Secondly, the noise in the signal may be correlated with an unknown distribution. Finally it may not be possible for the linear components to form the original signal.

The algorithm has already been formally introduced in chapter 3, with its pseudo code for implementation in code listing 1.

Reconstruction error is simply defined by the Frobenius norm of the PARAFAC representation as a percentage change of the original tensor. This is shown in equation 4.7

$$E_{PARAFAC}(R) = \frac{\|\boldsymbol{\chi} - \sum_{i=1}^{R} \boldsymbol{a}_i \circ \boldsymbol{b}_i \circ \boldsymbol{c}_i\|}{\|\boldsymbol{\chi}\|} \tag{4.7}$$

PARAFAC2

Motivation: The requirement for PARAFAC2 stems from a practical issue in measuring EEG, moving temporal sources in the assumed synchronous channels. Apart from noise there are various biological reasons for these time shifts to occur - some of which may be brain activity, sweating or muscle tension. These slow time shifts can change the zero level across channels considerably, and are difficult to prevent due to the recordings occurring at high frequency from different spatial locations. The time shift biases lead to components that are not rank one, making PARAFAC an inappropriate model. The fundamental limitation in CPD and ICA is that the signals must vary amongst channels by a scale factor, which does not allow for time shifts to be taken into consideration.

Consider again the CPD decomposition.

$$\begin{aligned} (\boldsymbol{\mathcal{Y}}_i)_{channel\ p} &= \boldsymbol{a_i} \cdot c_{i,p} \cdot \boldsymbol{t}_{\boldsymbol{p,i}}^T \\ \therefore \boldsymbol{\chi}_{channel\ p} &= \boldsymbol{A} \cdot diag(s_k) \cdot \boldsymbol{T_k} \end{aligned} \tag{4.8}$$

The decomposition is only unique following Harshman constraints shown in equation 4.9. The relative shifts must also be constant between the components.

$$\boldsymbol{T}_{\boldsymbol{k}}^T \cdot \boldsymbol{T_k} = \boldsymbol{H} \in \mathbb{R}^{R \times R} \tag{4.9}$$

This imposes the constraint that the correlation matrix T_k is to be independent of channel k.

Definition: Therefore the PARAFAC2 model is stated as follows:

$$\hat{\boldsymbol{\chi}} = \sum_{i=1}^{R} \boldsymbol{a_i} \circ (\boldsymbol{F}_i \cdot diag(\boldsymbol{c}_i) \tag{4.10}$$

Where $F_i = [t_{1,i}, t_{2,i} ... t_{N_c,i}] \in \mathbb{R}^{N_t \times N_c}$ consists of the time signatures for the n^{th} component. Note the matrix $\boldsymbol{F}_i \cdot diag(\boldsymbol{c}_i)$ can be interpreted as time-varying channel signature, thus providing information of the evolution of channel signatures over time. One important consequence of this is that the components of PARAFAC2 are exact for the temporal location, whilst for PARAFAC the component at a time may not correspond to the exact location in time. Therefore the EEG decomposition is more interpretable as well.

Reconstruction error can be written as shown in equation 4.11

$$E_{PARAFAC2}(R) = \frac{\|\boldsymbol{\chi} - \sum_{i=1}^{R} \boldsymbol{a_i} \circ (\boldsymbol{F}_i \cdot diag(\boldsymbol{c}_i)\|}{\|\boldsymbol{\chi}\|} \tag{4.11}$$

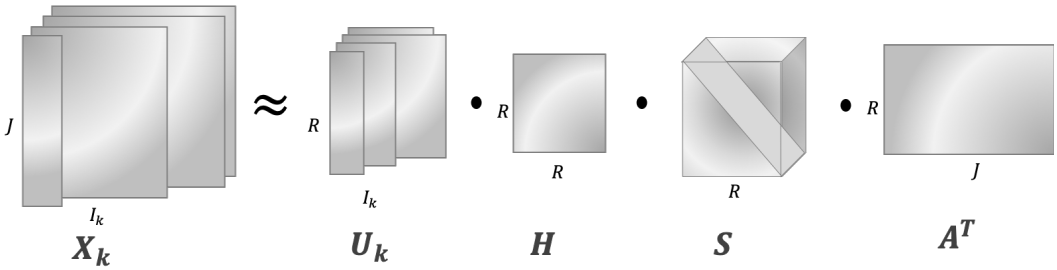

Figure 4.3: Visual representation of the Parafac2 decomposition.

This study also makes use of a real dataset with the specific reason of testing Parafac2, that is introduced formally in chapter 5. A drift is a long-lasting change of the DC potential, and therefore cannot be corrected through averaging using epochs. This means that they should be observable in scalograms. Figure 4.5 clearly demonstrates undesirable baseline drift artifacts. It was found that it, in fact, is superimposed to slower neural events. As the drifts can be of any nature (linear, non-linear, static or dynamic), no universal methods exists to reduce their effects - though analysis of variance (ANOVA) and wavelet based methods have achieved some success [55] [56].

The pseudo code is shown in listing 5, and has been implemented into the associated open source library with this project [3]. To the best of the author's knowledge, no other open source implementation of Parafac2 can be found.

Reconstruction error itself is sufficient to show that PARAFAC2 is able to better represent an EEG signal. This is tested in Chapters 6 and 7. The application to artifact removal is discussed in chapter 6.

Algorithm 5: PARAFAC2

Data: Matrices $\boldsymbol{X}_k \in \mathbb{R}^{I_k \times J} \forall k = 1...K$, rank R
Result: $\boldsymbol{U}_k \in \mathbb{R}^{I_k \times R}, \mathcal{S} \in \mathbb{R}^{R \times R \times K}, \boldsymbol{V} \in \mathbb{R}^{J \times R}$ and $\boldsymbol{\mathcal{H}} \in \mathbb{R}^{R \times R}$

Initialise $\boldsymbol{H}, \boldsymbol{S}[:,:,k] \leftarrow \boldsymbol{I}$
Initialise $\boldsymbol{V} \leftarrow$ using R eigenvectors of $\sum_{k=1}^{K} \boldsymbol{X}_k^T \boldsymbol{X}_k$ or randomly
Initialise $\boldsymbol{U}_k \leftarrow$ randomly
repeat
  for $n \leftarrow 1$ ***to*** N **do**
    $[\boldsymbol{P}_n, \Sigma_n, \boldsymbol{Q}_n] \leftarrow$ SVD with R components of $\boldsymbol{H}\boldsymbol{S}_{\boldsymbol{K}}\boldsymbol{V}^{\boldsymbol{T}}\ \boldsymbol{X}_{\boldsymbol{n}}^{\boldsymbol{T}}$
    $\boldsymbol{U}_n \leftarrow \boldsymbol{Q}_n \boldsymbol{P}_n^T$
  end
  for $n \leftarrow 1$ ***to*** N **do**
    $\mathcal{Y}[:,:,n] = \boldsymbol{U}_n^T \boldsymbol{X}_n$
  end
  $\boldsymbol{H}, \boldsymbol{V}, \hat{\boldsymbol{S}} =$ CPD_ALS($\mathcal{Y}$) for one iteration
  for $n \leftarrow 1$ ***to*** N **do**
    $\mathcal{S}[:,:,n] = Diag(\hat{\boldsymbol{S}}[\boldsymbol{n},:])$
  end
until *convergence criteria met*;

4.5 Family of Tucker Decomposition

Tucker decomposition, relative to Parafac, is more flexible (due to the higher order core array) but it is generally non-unique. Due to this property, Tucker in general requires fewer components than Parafac to be able to extract the features of interest. A simple example is shown in Figure 4.5.

For Tucker, one required information in each mode except the sample mode during feature selection. Due to the way a Tucker decomposition is performed (see section 9), its decomposition results are less intuitive to understand than that of CPD. In CPD, an observed artifact in one signature (temporal for example) can be easily traced in a different signature (frequency for example). Thereby an artifact can be appropriately defined as a rank one tensor. In Tucker the components in one mode can interact with other modes in a different component, due to the full core array. The lack of uniqueness will also mean the spatial topography is not taken into consideration, as the model can only distinguish component matrices to a certain rotation.

Recall that singular value decomposition provides a sum of rank one matrices, as shown in equation 4.12. One interpretation of this is the division of power into n components of $\boldsymbol{A}$. The method can be used for artifact removal in the manner described in section 6.4, but not artifact extraction as PARAFAC enables. An important point to note is that Tucker, as SVD for matrices, organises itself by an

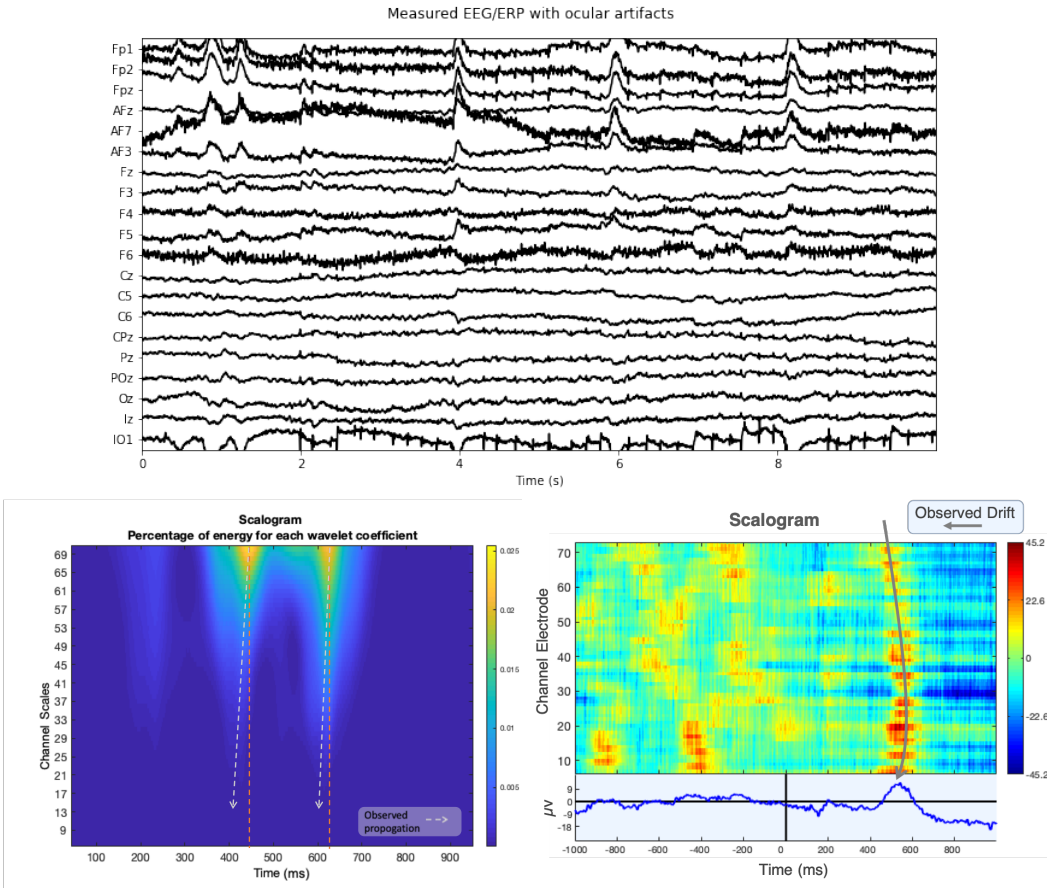

Figure 4.4: (a) A time series view of the channels from the real reading dataset. Many electrodes, such as AF7, experience baseline drifts. (b) Generated scalogram of the dataset. The arrowed lines indicate that there is a latency between the response of the occipital channels (nearer to the eyes, which respond first) compared to the frontal ones.

order of captured variance by each component.

$$\begin{aligned} \boldsymbol{A} &= \sum_{k=1}^{r} \boldsymbol{u}_k \sigma_k \boldsymbol{v}_k^T \\ \boldsymbol{A}_n &= \sum_{k=1}^{n} \boldsymbol{u}_k \sigma_k \boldsymbol{v}_k^T \\ \|\boldsymbol{A}\|_F^2 &= \sum_{k=1}^{r} \sigma_k^2 \end{aligned} \tag{4.12}$$

4.6 Data Fusion methods

The tensor factorisations discussed can be used for analysing matrices of EEG data, as the models decompose into coupled matrices that are analysed simultaneously. Couple Matrix Tensor Factorisation model (CMTF) allows coupled analysis of het-

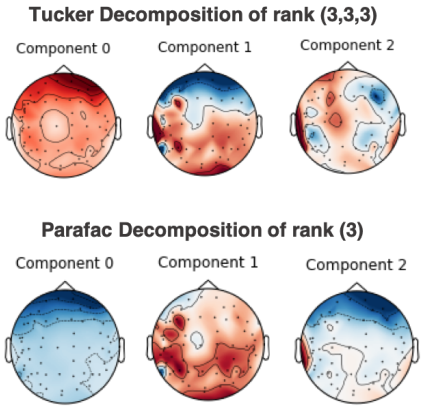

Figure 4.5: Parafac and Tucker decompositions with a very small number of components. This example, conducted on a real (MNE) dataset, illustrates that Tucker requires fewer components than Parafac to extract blinks.

erogeneous data, which decomposition methods cannot directly. In this study simultaneous MEG, EEG dataset have been used for further analysis. It should be noted that an entirely data driven approach is considered, and model driven approaches built on Bayesian statistics are out of the scope of this project.

Decomposition will provide spatial signatures $\boldsymbol{M_G}$, temporal signatures $\boldsymbol{T_V}$ and spectral $\boldsymbol{F_V}$. The MEG data will be decomposed into only spatial $\boldsymbol{M}$ and temporal signatures $\boldsymbol{T}$. Another possible dataset to fuse with could have been fMRI, which measures oxygenated blood flow. Most notably fMRI and EEG has previously been used by Acar et. al to identify activity patterns in Schizophrenia [57].

5 — Quantifying performance for artifact removal

Artifacts are of great interest for two reasons. Firstly they form a large proportion of the power of the signal, and impact numerous processing techniques negatively. However many neural activities may also be correlated with certain artifacts, such as between blink-evoked activity and other neural activity. Ocular artifacts are often the most prevalent and have the large power value, and will therefore be the most interesting for this project.

There are principally three categories of solutions that address these issues with ocular artifacts. The first method being to record with eyes closed, however this places a limitation on the activity and does not guarantee complete removal of blink activity. Rejecting trials or segments of datasets that may contain ocular activity is another simple method to address this problem. This involves rejecting large amounts of data that may not be easily reproducible. The third class of solutions involves correcting EOG activity. This is the category that this project addresses.

5.1 The problem set

To be able to measure the true success of the methods that have been implemented, it is important to have defined a ground truth. In this application the ground truth would be the perfect separation of neural activity and artifacts. Please note that artifact removal in this study is simply treated as a method of activity localisation. One way of doing this is to generate synthetic EEG data, allowing greater control over the spectral and spatial modes as well, which involves solving the forward problem for projection onto the scalp. Therefore simulating EEG is not trivial. The functions available from an existing library [58] are used in the manner shown in figure 5.1 to generate this dataset. A framework for artifact removal will be developed, and an open source dataset is used to test the tensor methods in an existing framework.

One method of compressing/localising blink artifacts would be to regress out reference signals near the eyes (channels such as FP_1 and FP_2). However this is practically difficult as it requires clean EOG channels, and artifact signals may propagate to EOG sites. Further they may also compress neural activity that is found common in the reference and frontal electrodes.

For all datasets, the sensor positions are defined as shown in figure 5.1.

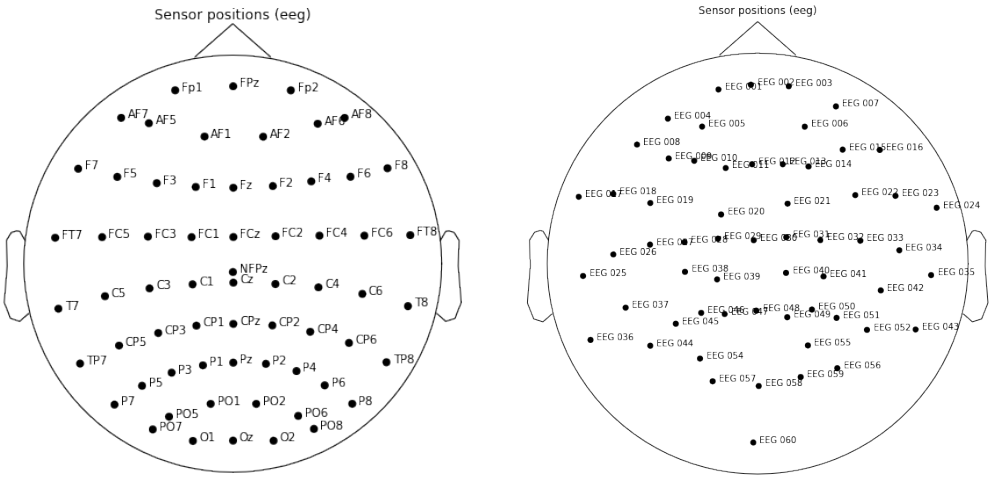

Figure 5.1: Topological view of the scalp Electrode positions described by the 10-20 system (left) and a synthetic version (right).

5.2 Overview of the real Datasets

The artifact removal methods were also tested on two real datasets, one being affected by drift frequencies and other noise sources more than the other. A summary of the datasets is found in table 5.1. MNE provide a simultaneously recorded EEG and MEG dataset [58], which has been used for this project. Simple experiments of auditory and visual stimuli were used to measure induced responses. Bad channels have also been removed manually. The dataset was readily filtered from 0 to 40 Hz, removing line noise and its associated harmonics as well as some other noise elements.

The less well behaved dataset is obtained from [59], and was measured in relation to the work by Henderson et al. [60]. It is a natural reading task, where the subject is expected to respond to the appearance of an animal's name. The dataset had some ground truth available as blink locations could be obtained from the separately recorded eye movements. The context of the measurements is not significant in this case, as long as other neural activities are present that can be recovered.

General description of the EEG dataset			
Property	Description	MNE Data	Reading data
Sampling Frequency f_s	Defines the frequency resolution	$150Hz$	$512Hz$
Number of electrodes	Defines the frequency resolution	$376\,(GRAD: 204, MAG: 102, STIM: 9, EEG: 60, EOG: 1)$	$72EEG$
Filtering properties	Sinusoid of fixed amplitude for each dipole source, synchronised across the layers	$(0.1, 40)Hz$	$(0.1, 100)$
Time Duration	The signal duration	$277.7s$	$102s$
Events	Defines the frequency resolution	Visual / Auditory events (x6)	reading and interpreting (x5)

Table 5.1: Properties of measured EEG datasets used

Generating Synthetic Data

A linear EEG model is assumed in most of the project, following the ideas of Parra et al. [61]. The source potentials are assumed to be additive $s(t)$, such that the signal can be easily represented from the model of a single current source $x(t) = \boldsymbol{a}s(t)$, as shown in equation 5.1 with an additive noise term.

$$\boldsymbol{X}(t) = \boldsymbol{A}\boldsymbol{s}(t) + \boldsymbol{n}(t) \tag{5.1}$$

$\boldsymbol{A}$ is referred to as the forward model. A similar equation may be obtained that maps the sensor activity to the sources. This is shown in equation 5.2, where V is referred to as the backward model. This is the equation of interest for generating synthetic EEG.

$$\hat{\boldsymbol{S}}(t) = \boldsymbol{V}^T\boldsymbol{x}(t) \tag{5.2}$$

One way to select $\boldsymbol{V}$ is to simply consider the least mean squares estimator, minimising noise $min_v \sum_t \|n(t)\|^2$. This has the solution $A^\dagger$. If noise covariance is known/estimated, a better distribution of true noise is known which can be incorporated in the solution. This is shown in equation 5.3.

$$\begin{aligned} \hat{\boldsymbol{v}}^T &= (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T \\ &= A^\dagger \end{aligned}$$

or with noise (5.3)

$$\begin{aligned} \hat{\boldsymbol{v}}^T &= (\boldsymbol{A}^T\boldsymbol{R}_n^{-1}\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{R}_n^{-1} \\ \therefore \hat{\boldsymbol{S}}(t) &= \boldsymbol{S}(t) + \boldsymbol{V}^\dagger n(t) \end{aligned}$$

The dataset was generated using MNE [58] with the parameters as tabulated in 5.2. A description of the generation process can be found in Figure 5.3. For the majority of the project, a simple dataset of EOG (blink) artifacts solved using the forward problem, and Gaussian noise was used. To generate the multivariate Gaussian distribution, a noise covariance were used from an actual recording that is available in MNE's dataset [58].

All significant properties for the purpose of this study from the real dataset have been preserved. This includes an almost perfect power spectral density, and ocular artifacts that have a rapidly decreasing amplitude with distance from the eyes. Effects from EEG rythms were also added for exploratory purposes.

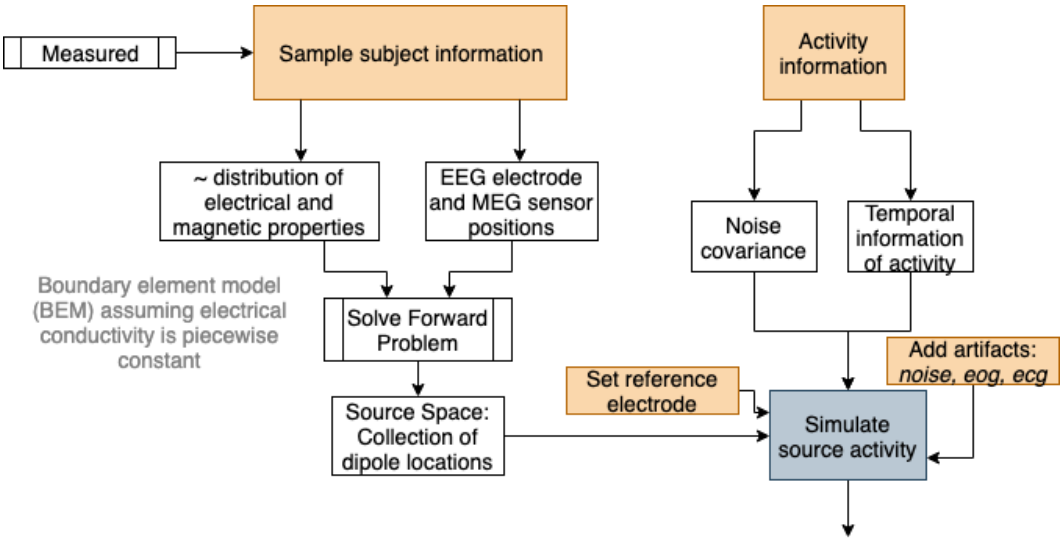

Figure 5.2: Simulation process of synthetic EEG data generation using the forward model.

The time series representation can be found in figure 5.3

5.3 Methods of quantification

Particular components of the specific dataset are extracted, therefore there is no meaning to splitting into train/validation/test. Reconstruction errors show how

General description of the EEG dataset		
Property	Description	Value
Sampling Frequency f_s	Defines the frequency resolution	$1100Hz$
Sinusoid harmonic f	Sinusoid of fixed amplitude for each dipole source, synchronised across the layers	$10Hz$
Time Duration	The signal duration	$10s$
Event/Epoch duration	The time an event (blink) will last	2
IIR Filter Coefficients	The denominator coefficients	[0.2, -0.2, 0.04]
Number of dipoles	A dipole is the location of a source	4

Table 5.2: Properties of the synthetic EEG dataset

well a model is able to represent the EEG signal, and while this will be taken into consideration, it by no means quantifies performance on source localisation/extraction.

Case A: Known ground truth

In this experiment it is possible to separate the noise, EOG(blink) and ECG (rythms affecting EEG) artifacts and therefore compare them to the results obtained. The correlation coefficient and relative mean squared error are apt metrics for this case, following equations 5.4 and 5.5. Note that there is no widely accepted definition for normalised/relative mean squared error. The fundamental issue here, however, is in the reconstruction of the EEG signal to allow such a comparison.

$$RMSE = \frac{MSE(\mathbf{X}_c - \hat{\mathbf{X}})}{MSE(\mathbf{X}_c)} \quad \text{where} \mathbf{X_c} \text{denotes the corresponding channel c for the reconstructed EEG signal.} \tag{5.4}$$

$$\rho_s = \frac{\left(vec(\hat{\mathbf{X}_c}) - \mu_{\hat{\mathbf{X}_c}}\right)^T \cdot \left(vec(\mathbf{X_c}) - \mu_{\mathbf{X}_c}\right)}{\|\left(vec(\hat{\mathbf{X}_c}) - \mu_{\hat{\mathbf{X}_c}}\right)\| \cdot \|\left(vec(\mathbf{X_c}) - \mu_{\mathbf{X}_c}\|\right)} \quad \text{where } \mathbf{X_c} \text{ denotes the corresponding channel c for the reconstructed EEG signal.} \tag{5.5}$$

True RMSE can only be calculated if the transform used, being CWT in this case, allows perfect reconstruction. Although with wavelets such as Morelet it is possible to achieve an almost accurate reconstruction, there is still an additional uncertainty when comparing the reconstructed signal to the true signal [62]. In order to overcome this uncertainty, a simple modification to the method is made.

$$\begin{aligned}
&\text{The wavelet transform is linear}\\
&\mathcal{W}[s(t)] = \mathcal{W}[as_1(t) + bs_2(t)] = a\mathcal{W}[s_1(t)] + b\mathcal{W}[s_2(t)]\\
&\text{As the noise is separable when the ground truth is known:}\\
&\mathcal{W}[as_{eog}(t) + bs_{noise}(t)] = a\mathcal{W}[s_{eog}(t)] + b\mathcal{W}[s_{noise}(t)]\\
&\text{But with the inverse, an approximation of the signals are constructed}\\
&\implies \mathcal{W}^{-1}\left[\mathcal{W}[as_{eog}(t) + bs_{noise}(t)]\right] = a\tilde{s}_{eog}(t) + b\tilde{s}_{noise}(t)
\end{aligned} \tag{5.6}$$

The artifact removal using decomposition gives a clean signal $s_{clean}(t)$ from $\tilde{s}(t)$ and not $s(t)$ (although it is a very good approximation of $s(t)$) without $s_{noise}(t)$. Let this be denoted as $\hat{s}(t)$. In this study $\hat{s}(t)$ will be compared to $\tilde{s}(t)$ instead of $s(t)$.

Consider the limitations of the Relative Root Mean Squared Error (RRMSE) in the context of EEG. Signals of a small amplitude will tend to give better RRMSE, due to a smaller denominator in the defined equation. As neural activity is of much smaller amplitudes than the artifacts, removing too many components (which includes important brain signals) may actually lead to an improvement in the metric. Mean Absolute Error (MAE) is a more robust version of Root Mean Squared Error. As the variance increases, the error magnitudes will increase. This is depicted in table 5.3. A more complete discussion comparing providing reasons to use the MAE can be found in [63]. Although the results of the study should be noted, it does not necessarily mean MAE is more appropriate for this project as only one aspect of the error characteristics are explored. The ground truth that is being compared to is normally distributed noise, for which RMSE provides a more accurate measurement (closer to the real), as errors will be Gaussian [64]. Note that the L-2 norm is scaled RMSE. Therefore relative RMSE will be used as the distance metric.

$$S = \frac{1}{n}\sum_{i=1}^{n}|e_i| \tag{5.7}$$

The correlation coefficient measures the cosine angle relative to the centroid. Geometrically if two points lie on the same radius from the centroid will have a coefficient of 1, and 0 if they form a right angle from the centroid. It is not found to be representative of the true performance if the reconstructed errors are larger than

Error metrics		
Metric	Low Variance	High Variance
R_e0	3	7
R_e1	3	3
R_e2	3	2
R_e3	3	0
MAE	3	3
RMSE	3	3.94

Table 5.3: Example showing how the variance of a signal may effect the RRMSE. R_ei denotes the i^{th} relative error.

the original signal. This will give an unusually high correlation coefficient. Since the strengths of RRMSE complement the weaknesses of the correlation coefficient as a performance metric, both will be reported to measure performance.

Case B: Unknown ground truth

A decomposition method's fit may simply be determined by the reconstructed error / Mean Squared Error (MSE) (refer to equations 4.7, 4.11).

There is no established method of quantification in artifact correction research that can be employed as a method of evaluating performance. To develop an estimated measure of performance, consider a scenario in which the artifact signal and neural activity signal are separated perfectly. Upon the assumption that the artifacts are not correlated to the neural activity, one would assume that a good separation should tend towards zero correlation. Based on this idea, it is suggested that the correlation between the reconstructed/clean signal with the original signal be used as a metric for performance. Although the blink artifacts decrease in power (approximately by the inverse square law), it is expected that electrodes closer to the ocular region will have a smaller correlation to those further away - under the further assumption that an individual artifact lasts for the same time duration in all signals.

The use of simple variability measures, such as standard deviation or mean absolute deviation, does not provide information about the underlying rythms. Therefore other methods are needed to obtain biologically useful information. Prior to discussing such metrics, it should be noted that analysis in the spectral domain is affected more by artifacts than in the temporal domain.

Entropy Measures Estimations of entropy allows one to quantify the irregularities in a time series. In the context of EEG two broad categories that may be explored include: spectral entropy and embeddings entropy [47]. Spectral entropies use the

power spectrum as their probabilities, whilst embedding entropies use the signal directly.

For spectral entropies, Renyi's entropy is generally preferred over Shannon's entropy in EEG literature, both shown in equation 5.8 and equation 5.8 respectively.

Spectral entropy

$$S = \frac{\sum p_k log p_k}{log(N)} \tag{5.8}$$

where p_k is the amplitude at frequency k, and N is the number of frequencies.

$$H_m = \phi^{-1}\left(\sum_f p_f \phi(T(p_f))\right) \tag{5.9}$$

where ϕ is defined to be $2^{(1-\alpha)x}$.
Note: setting ϕ to x gives Shannon's entropy.

Sample entropy may also have been used.

The metrics discussed in this section for measured EEG datasets are not sufficient to measure the performance of EEG, but rather are necessary checks. They have been tested qualitatively, an example of which is shown in Appendix A. Consider the procedure of artifact removal, the assessment of which components correspond to blinks is a fundamental problem that requires one to look at the results of the reconstructed signal. It is not known if a blink is a blink and not a different concentrated activity in the ocular region, and the best manner of determining this is by face validity of the correction. Therefore qualitative analysis of EEG signals will still be a very important factor in this study

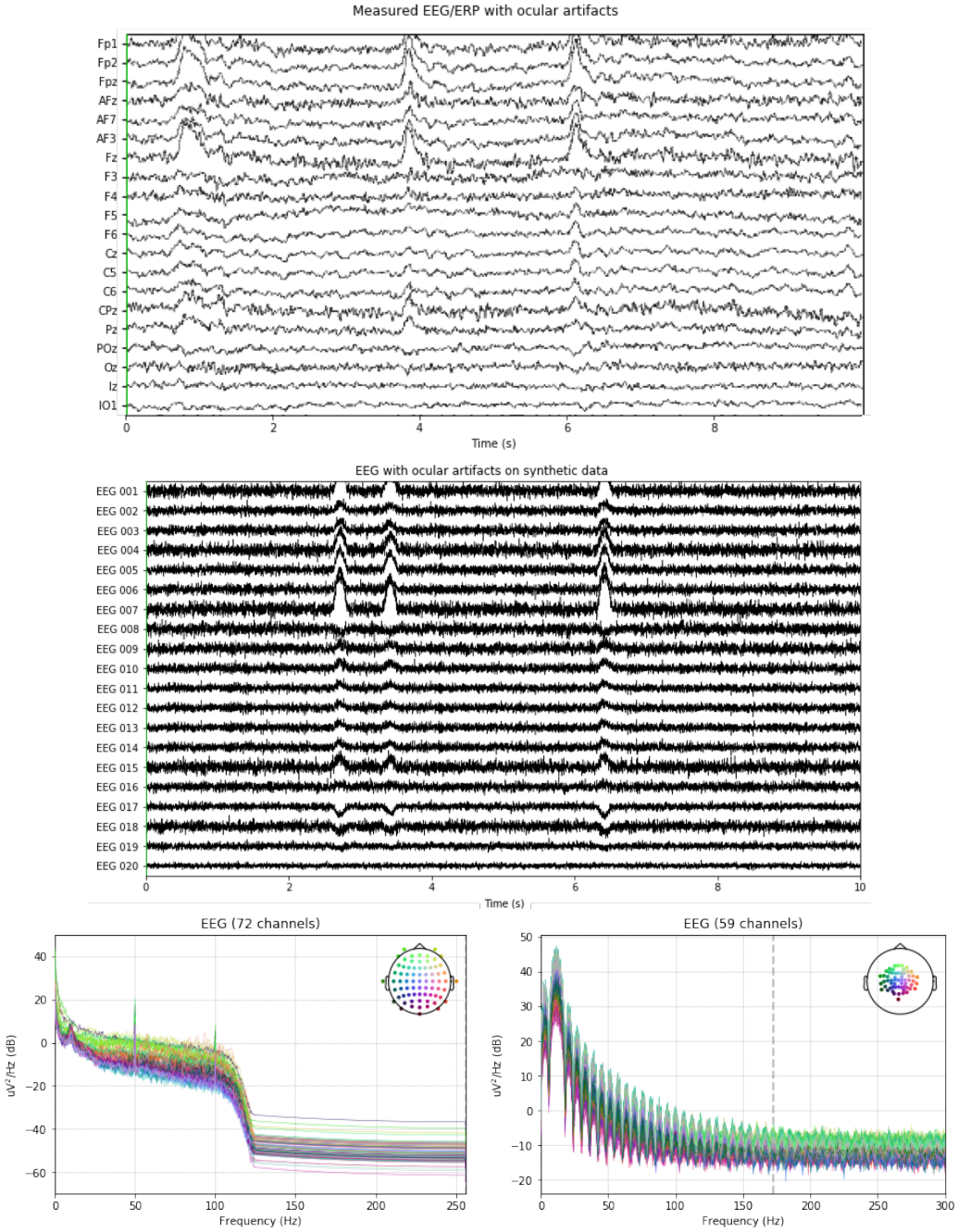

Figure 5.3: [Top]An incomplete view of the time domain signals of the generated measured and synthetic data respectively. The synthetic data was generated by solving the forward model and has three distinct blinks visible. It consists of EOG (blinks) and Gaussian noise only. [Bottom] The power spectral density is $1/f$ of measured and generated data respectively. PSD generated using MNE.

6 — Applying decomposition for artifact removal on Synthetic Data

Previous approaches include simplifying the generation of the eye movement to assume one source. A detailed understanding of the eye movement voltages is not necessary for this project, however it is important to note that the measurements are of voltage differences are recorded. Therefore values may become negative at one side of the scalp. Electrooculogram (EOG), measurements of electrodes close to the eyes, are commonly used in ocular artifact removal. This will be available in Synthetic datasets, as well as one real.

Consider the general case of Fourier Transforms, including the STFT that may be appropriate for EEG signals. Here the signal is transformed to a spectral domain where it can have a magnitude and phase. Whilst it is possible to deal with only the power of the signals by taking the absolute value of the complex number: $||z \in \mathbb{C}||$, clearly information is will not allow for reconstruction.

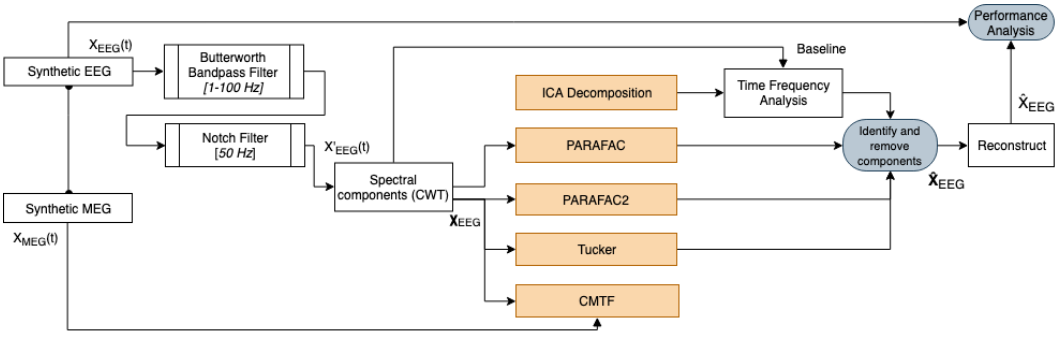

Figure 6.1: Artifact removal using decomposition methods

All methods were applied to 8 seconds of the data, consisting of three blinks (larger datasets were not possible due to performance).

6.1 Deriving a method for artifact removal

This section builds on the basic definitions and assumptions stated in the linear modelling of EEG in Section 5.2. The artifact removal method for tensors has been inspired from the two dimensional case derived by Parra et al. [61].

The linear combination of the surface potentials corresponding to the artifacts is as given in equation 6.1, assuming a linear estimate of the source model as shown in equation 6.2.

$$\boldsymbol{X}_{artifacts}(t) = \boldsymbol{A}\boldsymbol{S}(t) \tag{6.1}$$

$$\hat{\boldsymbol{S}}(t) = \boldsymbol{A}^T\boldsymbol{X}(t) \tag{6.2}$$

Here A depicts the coupling between source signals $x(t)$. To obtain a clean signal, simply subtract the artifact activity from the original. This is done by projecting the signal to the the normalised nullspace of A: $\boldsymbol{A}\boldsymbol{A}^\dagger$.

$$\begin{aligned} \boldsymbol{x}_\perp(t) &= \boldsymbol{x}(t) - \boldsymbol{x}_{artifact}(t) \\ &= (\boldsymbol{I} - \boldsymbol{A}\boldsymbol{A}^\dagger)\boldsymbol{x}(t) \end{aligned} \tag{6.3}$$

$\boldsymbol{A}^\dagger$ is the pseudo inverese of A, and is used as the separating matrix here. Using equation 6.3 a verification of the correlation metric as a means for quantifying performance can be seen. Here $\boldsymbol{x}_\perp$ is orthogonal to $\hat{\boldsymbol{S}}(t)$, which can be easily shown as in equation 6.4.

$$\boldsymbol{x}_\perp \cdot \hat{\boldsymbol{S}}(t) = (I - \hat{\boldsymbol{A}}\hat{\boldsymbol{A}}^\dagger)\boldsymbol{x}\hat{\boldsymbol{S}}^T = 0 \tag{6.4}$$

Note that this subspace projection will geometrically reduce the rank.

In the case of ICA, $\boldsymbol{y} = \boldsymbol{W}\boldsymbol{x}$ has components that are maximally independent, where W is the un-mixing matrix. Selecting components that correspond to artifacts from $y(t)$ as $\tilde{y}(t)$, the clean data is given by $\boldsymbol{x}_C = x(t) - A\tilde{y}(t) = x(t)[I - A\tilde{W}]$.

Now consider the Tucker representation of the EEG signal. To derive the equivalent, the mode product representation is considered. In a similar manner a projection onto the nullspace is used, however for a particular mode only. Equation 6.5 illustrates an example for mode 1.

$$\begin{aligned} \chi &= \kappa \times_1 \boldsymbol{U}^{(1)} \times_2 \boldsymbol{U}^{(2)} ... \times_N \boldsymbol{U}^{(N)} \\ \implies \chi_{clean} &= \kappa \times_1 \boldsymbol{U}^{(1)} (\boldsymbol{I} - \hat{\boldsymbol{A}}\hat{\boldsymbol{A}}^{\dagger}) \times_2 \boldsymbol{U}^{(2)} ... \times_N \boldsymbol{U}^{(N)} \end{aligned} \tag{6.5}$$

Therefore any particular mode can be reduced with this representation and the signal reconstructed.

A similar derivation of Parafac is obtained if it is written in the mode-product form shown in equation **??**. Note $\mathcal{I}$ is a super-diagonal tensor of ones.

$$\chi = \mathcal{I} \times_1 \boldsymbol{U}^{(1)} \times_2 \boldsymbol{U}^{(2)} ... \times_N \boldsymbol{U}^{(N)} \tag{6.6}$$

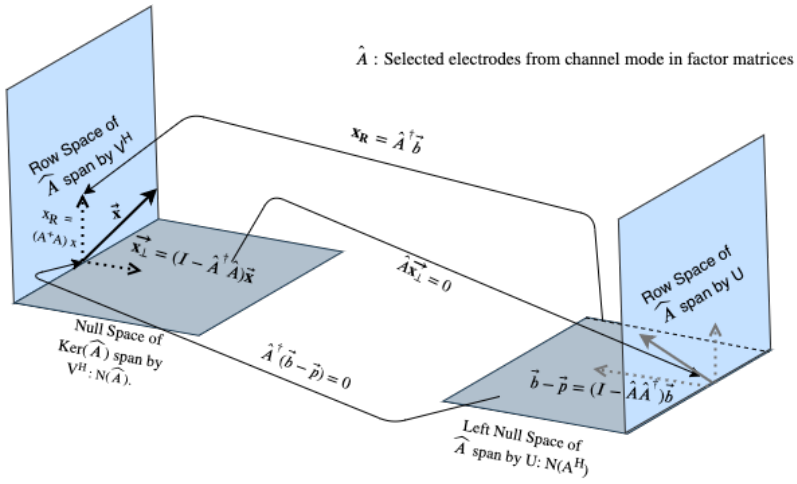

Figure 6.2: Illustration of projection onto the Null Space for artifact removal. The direction $\boldsymbol{x}_{\perp}$ allows reconstruction of a factor matrix on a subspace, such as the channel signature, with aritfact components removed.

Whitening is sometimes a suggested preprocessing step for general blind source separation methods to make the problem well posed.

6.2 Baseline ICA

When considering what the best method is, the question of whether the signals should be treated as non-linear deterministic or stochastic systems arises. There are reasons to believe both ways. As there is no notion of a pure signal, and it is not possible by any known means to determine its characteristics at any given point - it will be considered a stochastic system in this study. The objective is to remove the artifacts without effecting the quality of the EEG signal.

Consider the type of artifacts. Ocular artifacts can generally be seen in EEG, and more clearly recorded in EOG, due to their larger amplitudes relative to background activity. Muscle artifacts are much more varied in their characteristics,

as the amplitude and durations depend on what muscle was contracted and how. Cardiac artifacts have low amplitudes and are periodic in nature. It is expected that a well placed reference signal should make such noise insignificant.

There is no intent to provide an extensive survey of the matrix methods used for artifact removal in EEG signals, ICA is considered as the blind source separation method for the baseline model. The advantage of blind source separation methods is that they do not require a reference waveform. ICA solves the blind source separation problem explicitly, whilst other methods such as the empirical mode decomposition consider each channel separately. For further study on such methods, see It is expected that EMD will perform better than ICA on an ideal dataset - but not necessarily in the real world case. The technique decomposes a signal into intrinsic mode functions (IMFs), which are basis functions. An artifact, a mixture of the signals may be represented by one of these basis functions.

Implementation and Results

In this project, the fundamental ideas of the ICA method for EEG as proposed by Makeig et. al [65] are followed. There are three methods to perform ICA: *'fastica'*, *'infomax'* and *'picard'*. These were not heavily explored, and the most widely used method was selected: *'FastICA'* [66]. A brief summary is given below.

The fundamental idea behind FastICA is to find a direction $\boldsymbol{W}$ that maximises the negentropy of $\boldsymbol{W}^T x$. The negentropy is a method of predicting nongaussanity of a dataset. A method of fixed point iteration is used to iteratively update $\boldsymbol{w}$, as shown in equation 6.7, until the convergence criterion is met.

$$\boldsymbol{w}^{n+1} = \frac{\mathbb{E}(\boldsymbol{x}g(\boldsymbol{w}^T\boldsymbol{x}) - \mathbb{E}(g^{'}(\boldsymbol{w}^T\boldsymbol{x}))\boldsymbol{w})}{||\mathbb{E}(\boldsymbol{x}g(\boldsymbol{w}^T\boldsymbol{x}) - \mathbb{E}(g^{'}(\boldsymbol{w}^T\boldsymbol{x}))\boldsymbol{w})||^2} \tag{6.7}$$

where $\boldsymbol{x}$ is the input, prewhitened data

Using ICA decomposition, components that correspond to strong ocular activity were removed as shown in figure 6.3. Note that only the first twenty components are shown.

Removing such components, a reconstruction was performed: $\boldsymbol{W}^T X$. As the only component apart from EOG was noise for the generated synthetic data, the objective with any strategy would be to replicate this noise after cleaning. The results are shown in figure 6.4.

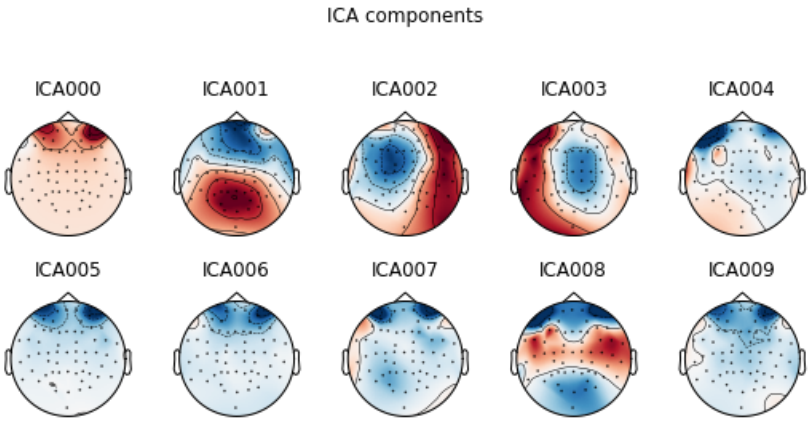

Figure 6.3: Visualisation of the first 10 components obtained using Independent Component Analysis. Components on Synthetic data. 0 represents Ocular artifacts strongly.

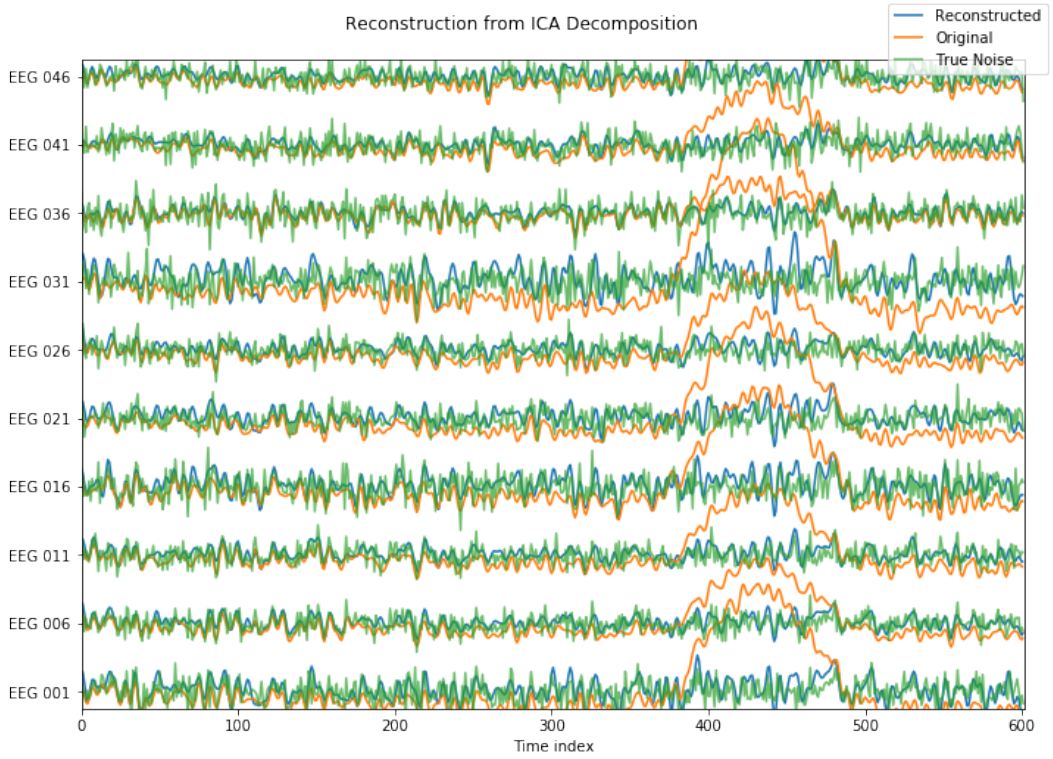

Figure 6.4: Reconstructed signals using ICA decomposition of ten electrodes. The reconstruction is replicating Gaussian noise as EOG artifacts are removed. The y axis is fixed from -20 to 20 μV has been removed for cleaner visualisation.

Analysis and Testing

In the case of synthetic data, the mean squared error and Pearson's correlation measured for each electrode are as shown in figure 6.5.

The mean relative mean squared error of all the channels: $\frac{1}{N}\sum_{c}^{\forall C}\frac{MSE(\mathbf{X}_c - \hat{\mathbf{X}})}{MSE(\mathbf{X}_c)}$ was

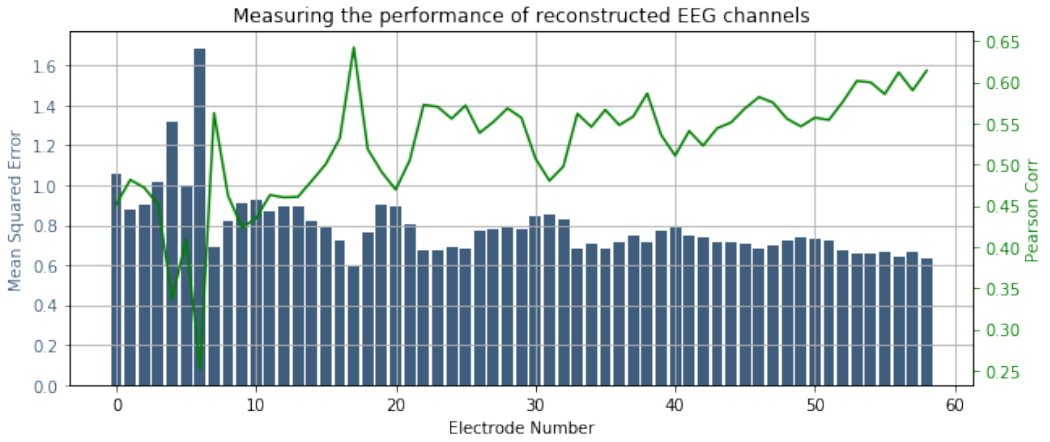

Figure 6.5: Root Mean Squared Error and Pearson's Correlation Coefficient as performance metrics for ICA.

found to be 0.79347.

6.3 Parafac family

Parafac and Parafac2 will have the same implementation scheme for artifact removal, as the result of both the tensor methods may be given by three factor matrices.

Implementation and Results

Building on the discussion of properties for EEG in section 4.4, and inspired by the work from [22], an artifact removal method was defined for the decomposition strategies (as shown in listing 6. $\boldsymbol{A, B, C}$ are the factor matrices defined to be the spatial, temporal signatures and spectral/scales. The algorithm employs a least squares approach estimates the activities corresponding to each component.

Algorithm 6: Artifact removal using the Tucker or Parafac family of decompositions

Data: $\chi \in \mathbb{R}^{I \times J \times K}$
Result: $\chi_{clean} \in \mathbb{R}^{I \times J \times K}$

1 Tucker/Parafac model on χ with sufficiently large components
2 Identify N components corresponding to artifacts
3 Form a matrix using these N components: $\tilde{\boldsymbol{A}} \in \mathbb{R}^{K \times N}$
4 Project onto the nullspace of $\tilde{\boldsymbol{A}}$. The columns will be the bases for $P_{M^\perp} = I - \tilde{\boldsymbol{A}}\tilde{\boldsymbol{A}}^\dagger$
5 The clean tensor can now be calculated as the projection of χ onto $M^\perp$: $\boldsymbol{\chi_{clean}} = \boldsymbol{\chi} \times_3 \boldsymbol{P_{M^\perp}}$

Please note that the decomposition methods for artifact removal provide a generic

framework for all, so as long as the artifact can be numerically determined. Step 2 of the algorithm in Listing 6 requires a qualitative analysis. The factors are examined in the electrode mode and the temporal mode to determine what can be removed. This is illustrated in figure 6.6. The components that are suspected to be related to blink artifacts are removed, and the tensor reconstructed in accordance to steps 3, 4 and 5. An example of the results obtained from Parafac is found in figure 6.7 and figure 6.8, where the signals are represented in a temporal and spatial ddomain respectively.

The topological diagram comparing the real and clean signals of figure **??** shows, qualitatively, that Parafac2 has performed well. It was expected that it will provide better results than Parafac. The component and comparison with the clean signal in the temporal domain can be found in the appendix, as they add little to the discussion.

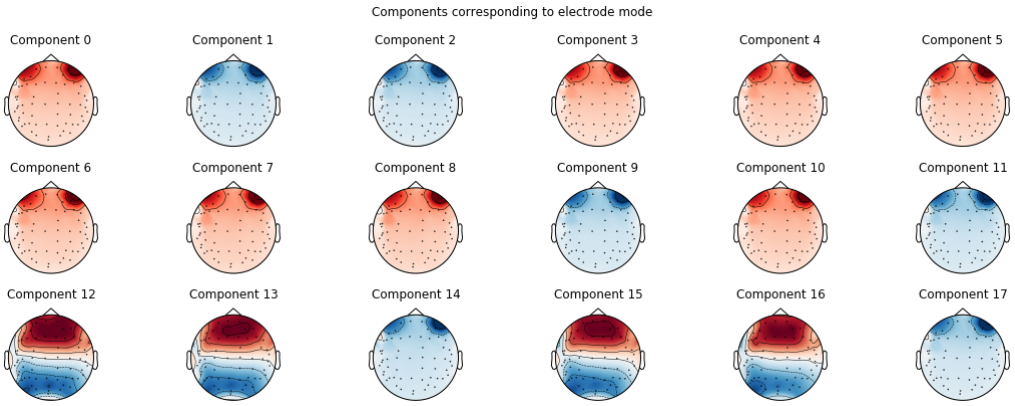

Figure 6.6: Visual representation of the components corresponding to the temporal mode of the Parafac decomposition. For illustrative purposes, a rank of 18 was selected (which corresponds to the number of components), with the analysis performed over a time interval of $\tilde{1}.82$ seconds or 2000 time points. From the diagram it can be seen that components 10 and 16 are the most likely candidates for detecting blink artifacts. This uses the generated synthetic data.

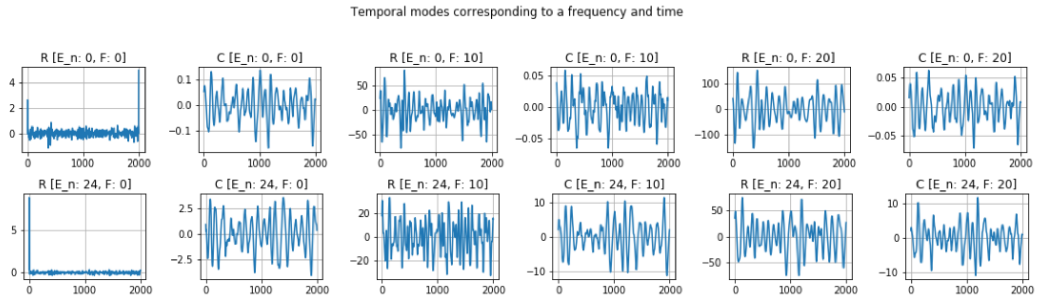

Figure 6.7: Real and Clean signals obtained from PARAFAC at certain frequencies, represented by R and C respectively in the diagram, represented in time. E_n corresponds to the electrode index, and F_n corresponds to the frequency component. Note the large differences in amplitude between R and C. These do not represent the time series, only its signatures.

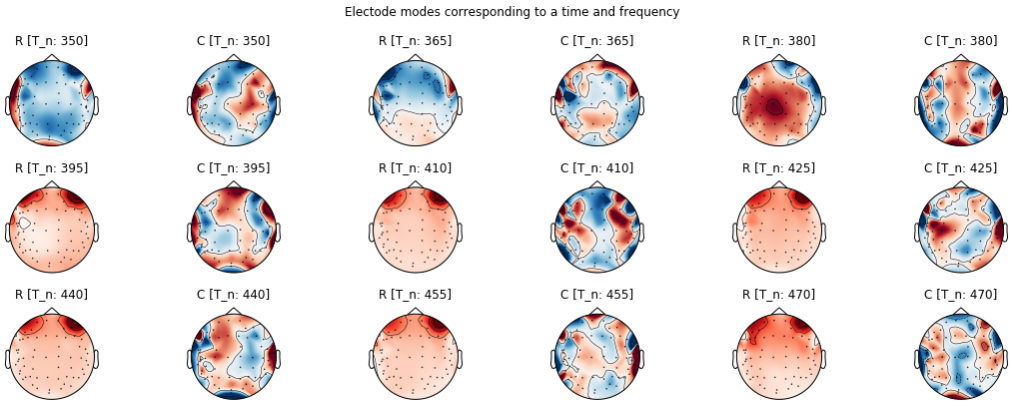

Figure 6.8: Real and Clean signals obtained from PARAFAC after reconstruction. T_n denotes the current time index. Note: a blink occurred at ≈ time index 380 (0.63 seconds), lasting for 110 indices (0.18 seconds) in the synthetic data.

Analysis and Testing

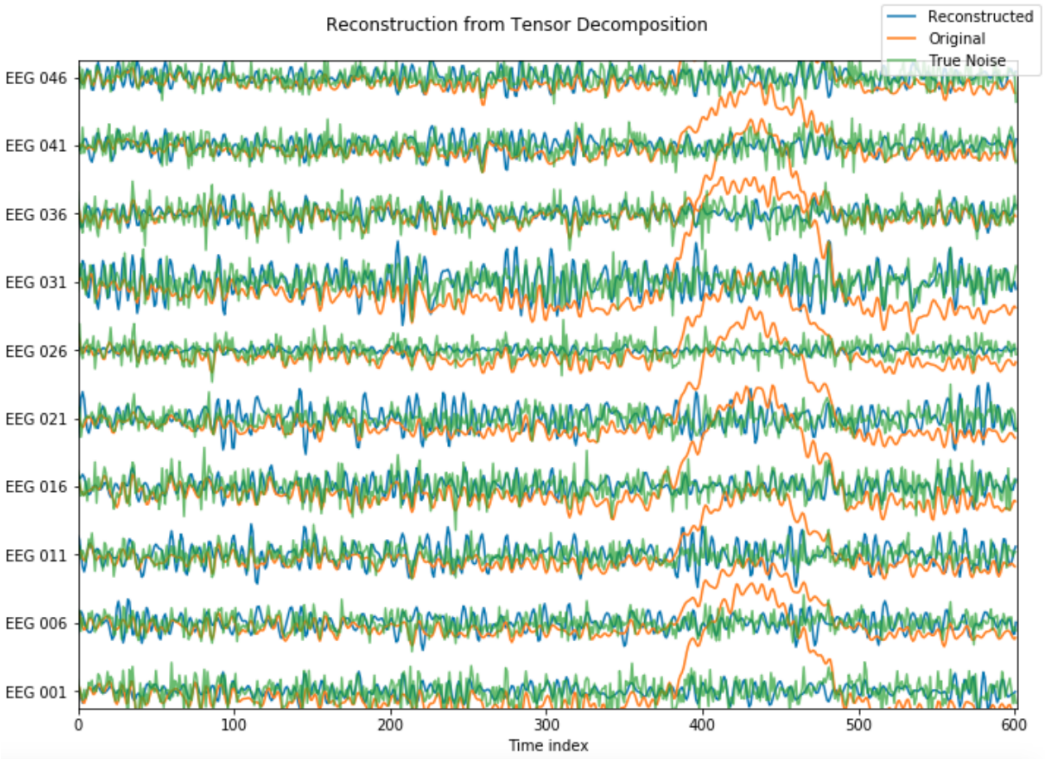

Figure 6.9: Reconstructed signals using Parafac decomposition. The reconstruction is replicating Gaussian noise as EOG artifacts are removed. The y axis is fixed from -10 to 10 μV has been removed for cleaner visualisation.

Clearly the amplitudes of figure 6.7, as well as the frequencies visible, show suppression of some components of the signal. From the topological view of figure 6.8, it can be seen that on numerous occasions it was blink artifacts that were suppressed. In a qualitative sense it seems the implementation has performed well.

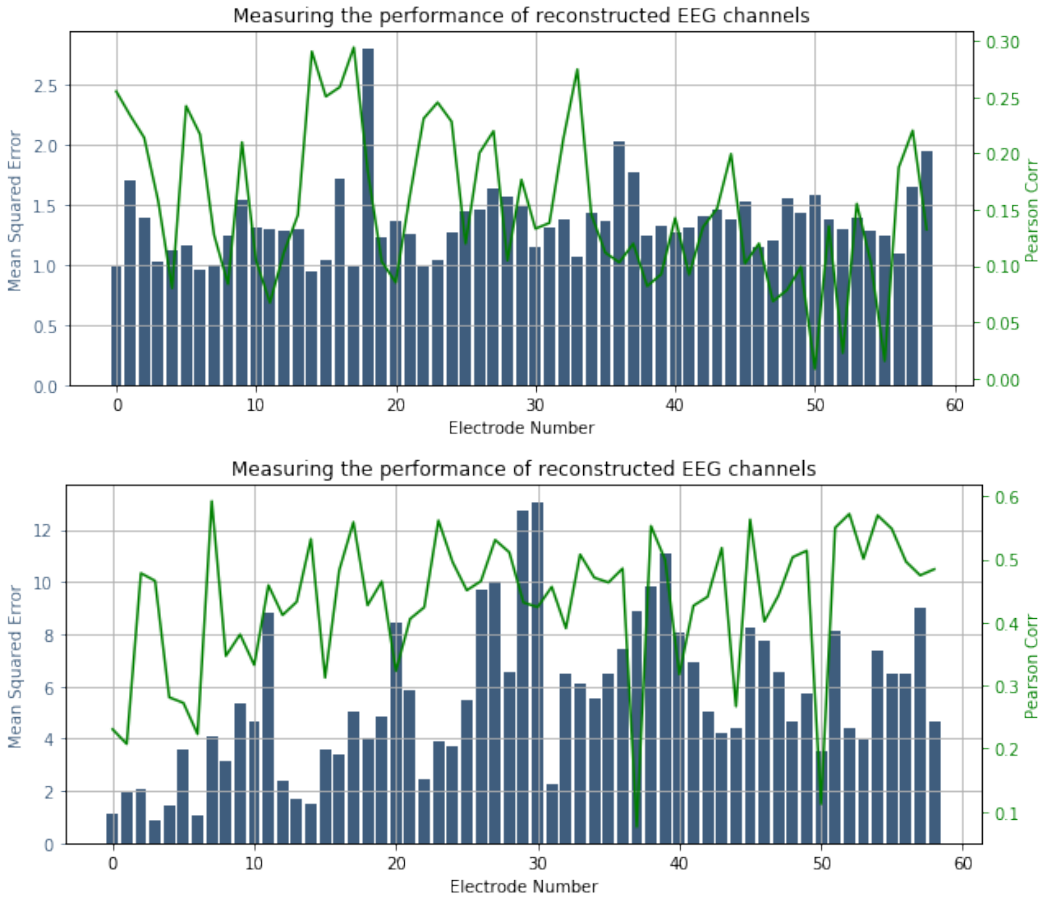

Figure 6.10: Relative Root Mean Squared Error and Pearson's Correlation Coefficient as performance metrics for (a) CPD (b) Parafac2.

The presence of true negatives (such as that shown in time index 380) suggests that blink artifacts may not have been the only extracted components.

The Parafac model, for the measured dataset, does not perform as well as had been hypothesised in Section 4.4. One reason for this could be that the electrodes contain correlated noise of an unknown distribution. The signals are non-stationary, as seen by the results of the Dickey-Fuller test in figure 6.11. All p-values were zero for the generated data, as was expected.

There is no lack of true synchronicity integrated into the synthetic data, because of which one would expect Parafac2 to perform only as well as CPD.

6.4 Tucker family

Implementation and Results

The artifact removal method for Tucker was introduced by Acar et. al [44]. As was the case for the PARAFAC family and ICA, the components of interest (that may

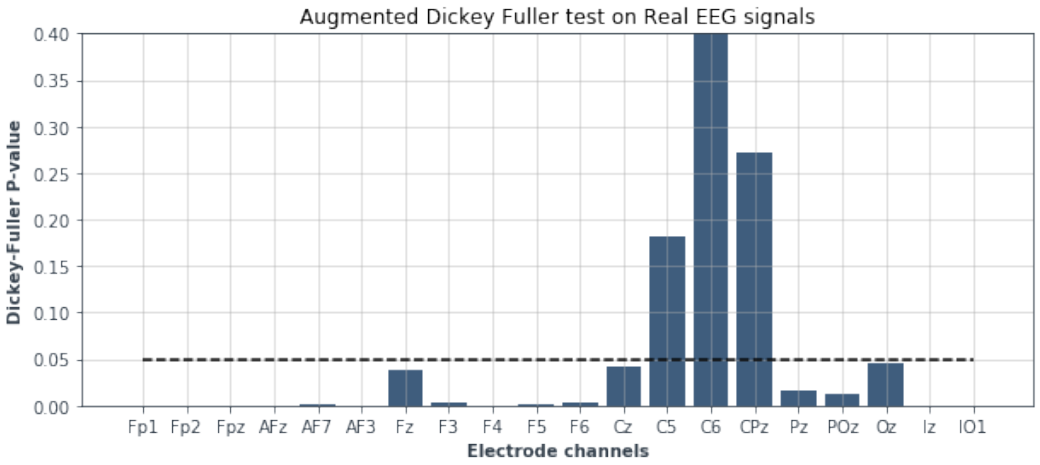

Figure 6.11: The computed p-values from the augmented Dickey-Fuller test. A line has been drawn for the null hypothesis rejection at a widely accepted value of 0.05. Note only the first 20 electrodes have been shown, and the y axis has been limited to 0.4 for illustration purposes.

correspond to artifacts) have to be identified and removed.

The results for the synthetic dataset can be viewed in figure 6.12.

Analysis and Testing

This approach has yielded similar success to PARAFAC In the case of the synthetic dataset, the relative mean squared error and Pearson correlation were measured as shown in figure 6.13. On both measurement metrics used, small improvements were seen compared to CPD. The representation does not perform as well as PARAFAC2.The mean relative mean squared error of all the channels was found to be 1.5723.

The lower entropy clearly shows denoising has been performed. However little assurance can be gained regarding the removal of blink artifacts, which was the initial objective. As a necessary check, rather than a sufficient one, the difference between the raw and clean entropy should be decreasing with an increasing index of electrodes. This is because the electrodes are laid out in the array such that the one closest to the eyes in the topogrophy of the head are found first. If the noise that has been removed is mostly blinking artifacts, a trend of larger to smaller deviation against the electrode index would be expected. This is observed as hypothesised in Figure 6.14.

For both entropies, the expected condition is met. Apart from the necessary sanity checks and qualitative analysis, no further quantitative method to determine whether the artifacts removed are indeed of the desired kind has been performed.

Further comparison of individual electrodes show that those closest to the ocular

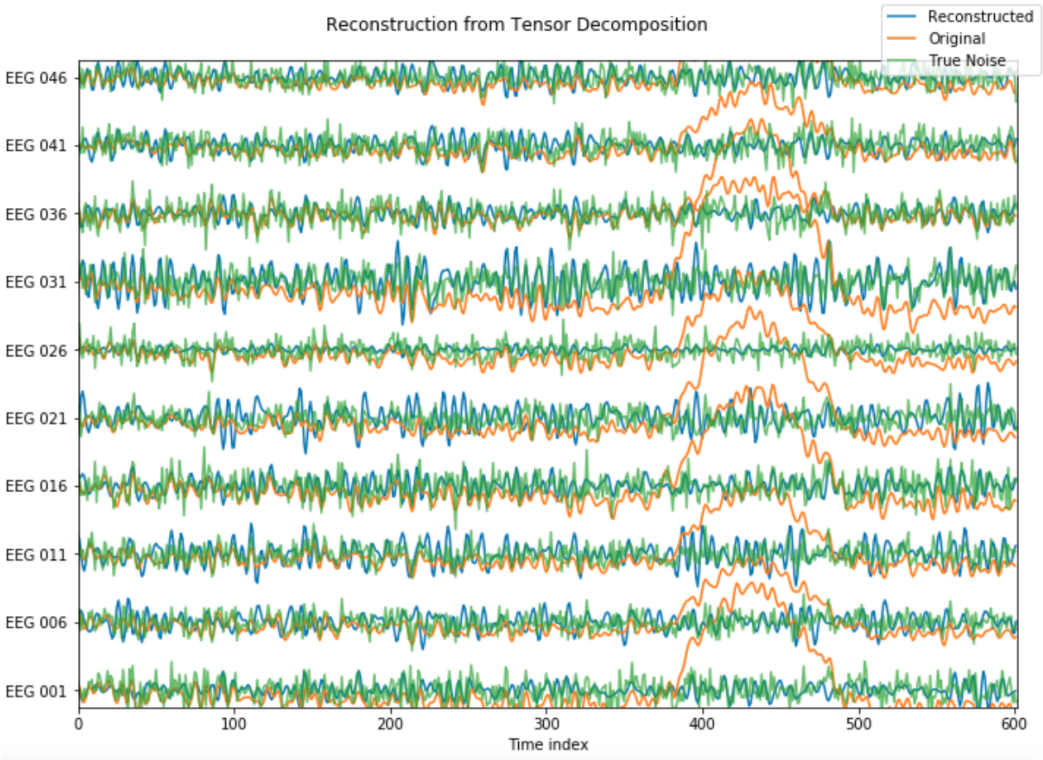

Figure 6.12: Reconstructed signals using Tucker decomposition of ten electrodes. The reconstruction is replicating Gaussian noise as EOG artifacts are removed. The y axis is fixed from -10 to 10 μV, and has been removed for cleaner visualisation.

activity were best recovered in the case of Tucker. This is, in fact, the opposite of what was observed in ICA. Both show better correlation for channels further away from the removed activity, perhaps due to a relatively smaller change from the removal of these components. This is not indifferent to what was expected, as the components corresponding to ocular artifacts were attempted to have been removed. It cannot be concluded that ICA performs better than Tucker, or vice versa, but this result does demonstrate that Tucker was able to localise activity better than ICA, though perhaps more components need to be observed.

6.5 Data Fusion

Implementation and Results

CMTF conducted by fusing EEG and MEG through CPD reconstructed the signals as shown in Figure 6.16 and 6.17. The overall RRMSE for the case of EEG was calculated as 1.662.

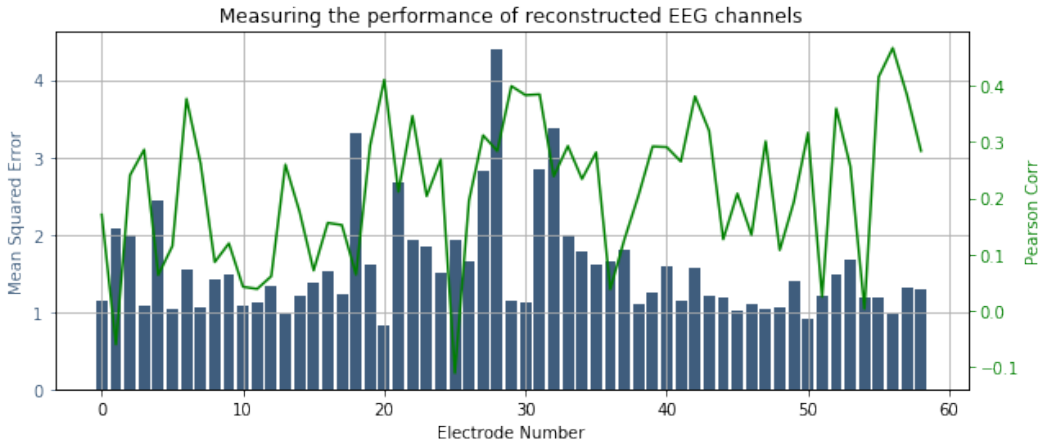

Figure 6.13: Root Mean Squared Error and Pearson's Correlation Coefficient as performance metrics for Tucker.

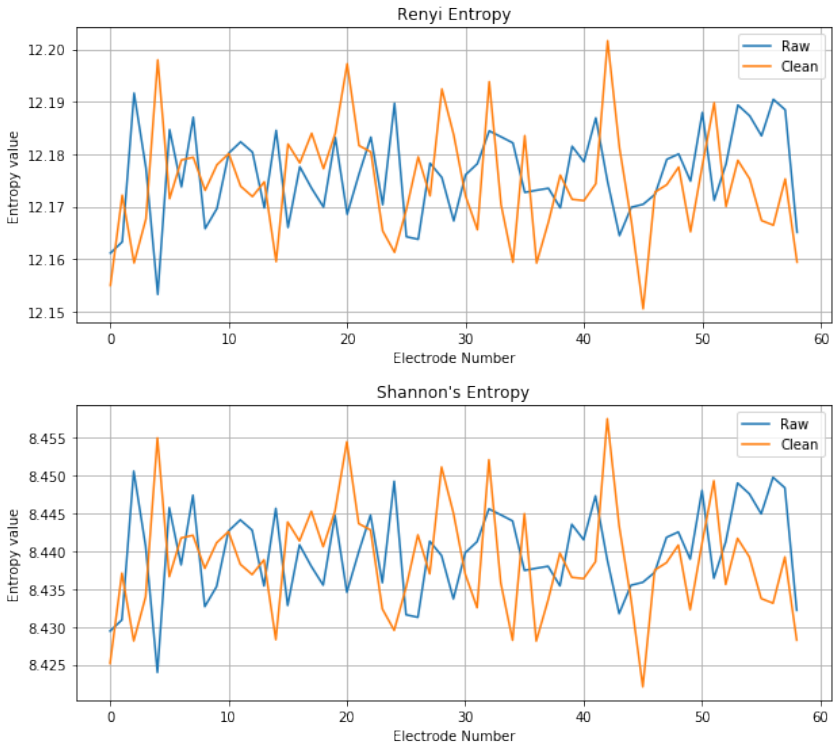

Figure 6.14: Entropy Measures

Analysis and Testing

MEG signals can record absolute neural activity, without the need for a reference, and does not have operational noise present. It was expected that CMTF will be able to achieve higher performance in extracting blink artifacts, due to two sources of information for localisation. Optimising the two signals simultaneously achieved a slightly worse performance. Interestingly most components in the elec-

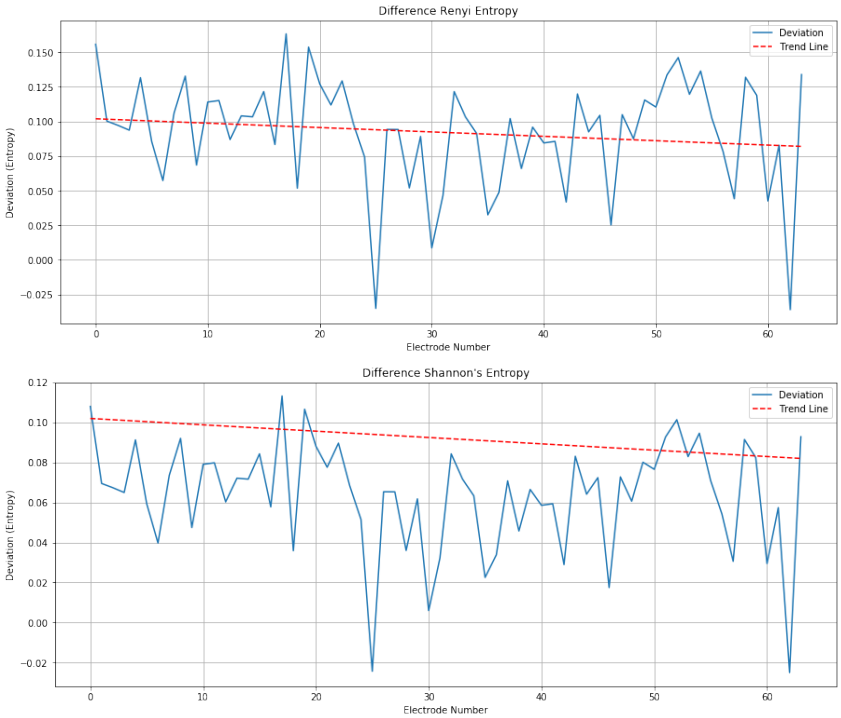

Figure 6.15: Entropy Measures

trode signature seemed to indicate blinks as shown in Figure 6.18.

6.6 Performance

CORCONDIA for CPD

To determine the appropriate number of components CORCONDIA , introduced by Bro et. al [67], was applied. The paper derives an appropriate method by considering the relationship between PARAFAC and TUCKER.

$$\begin{gathered} Parafac \implies \boldsymbol{\chi} = \boldsymbol{AT}(\boldsymbol{C} \otimes \boldsymbol{B})^T \\ \boldsymbol{A} \in \mathbb{R}^{I \times R}, \boldsymbol{B} \in \mathbb{R}^{J \times R}, \boldsymbol{C} \in \mathbb{R}^{K \times R}, \boldsymbol{T} \in \mathbb{R}^{R \times RR} \end{gathered} \tag{6.8}$$

This can be viewed as a restricted Tucker model with core T.

Fitting the Tucker model using components from PARAFAC, to determine if the components found from PARAFAC sufficiently describe Tucker.

$$\begin{gathered} \sigma(\boldsymbol{G}) = \|\boldsymbol{X} - \boldsymbol{AG}(\boldsymbol{C} \otimes \boldsymbol{B})^T\|^2 \\ \text{For optimal } \boldsymbol{G} \implies list(\boldsymbol{G}) = (\boldsymbol{C} \otimes \boldsymbol{B} \otimes \boldsymbol{A})^{\dagger} list(\boldsymbol{X}) \end{gathered} \tag{6.9}$$

Without any proofs it will be stated that a perfect fit for a PARAFAC model, the core tensor $\boldsymbol{G}$ must be a superdiagonal array of ones (the identity I matrix in the two

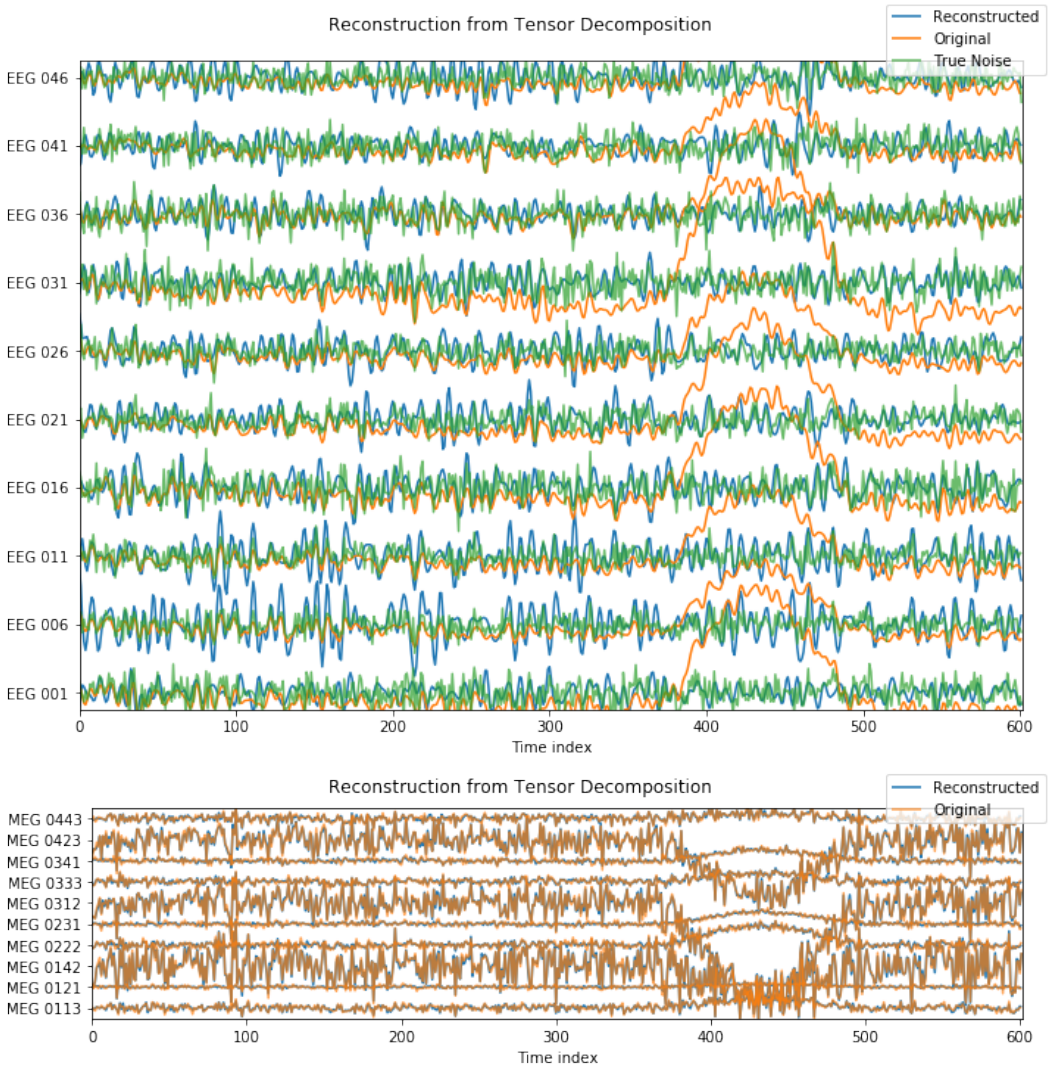

Figure 6.16: Reconstructed (a) EEG and (b) MEG signals using Coupled Matrix Tensor Factorisation of ten electrodes. The reconstruction is replicating Gaussian noise as EOG artifacts are removed. The y axis of the EEG signal is fixed from -10 to 10 μV, and -180 to 180 for the magnetic field in MEG. The y axis have been removed for cleaner visualisation.

dimensional case). CORCONDIA simply exploits this by measuring the similarity between the superdiagonal core $\boldsymbol{T}$ and least-squares fitted $\boldsymbol{G}$. In other words it is measuring the 'superdiagonality'.

$$\text{CORCONDIA} = 100\left(1 - \frac{\sum_{i=1}^{R}\sum_{j=1}^{R}\sum_{k=1}^{R}(g_{ijk} - t_{ijk})^2)}{R}\right) \tag{6.10}$$

The results from the core consistency analysis are shown in Figure 6.19. They suggest that, for the size of the dataset used, $\approx 24-32$ components for the Synthetic dataset, $27-32$ for the MNE dataset and $28-32$ for the real dataset are most ap-

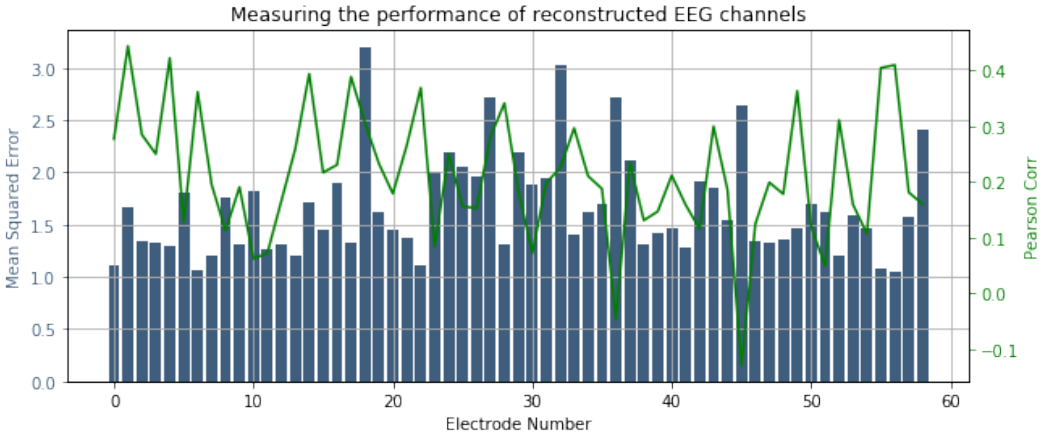

Figure 6.17: Root Mean Squared Error and Pearson's Correlation Coefficient as performance metrics for the EEG signal of CMTF.

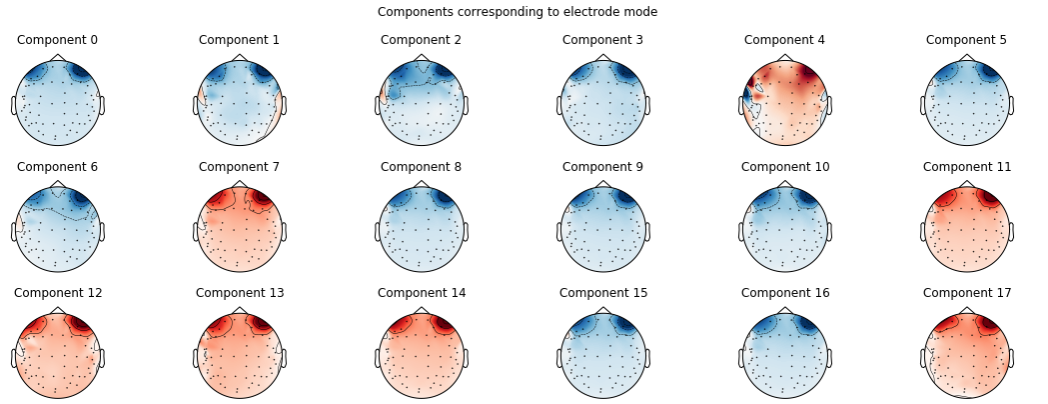

Figure 6.18: The first 18 components of the EEG channel signatures from CMTF of EEG and MEG data.

propriate.

Comparing performance

Considering the limits of the metrics, as discussed in chapter 5, clearly FastICA has achieved the best performance.

6.7 Optimisations for larger datasets

This section briefly discusses the significant computational optimisations, conducted during the course of this project. Although the results of the analysis do not affect the analysed performance of tensor methods, it is an important aspect of the project that determines the usefulness of the implementations in neuroscience in the foreseeable future.

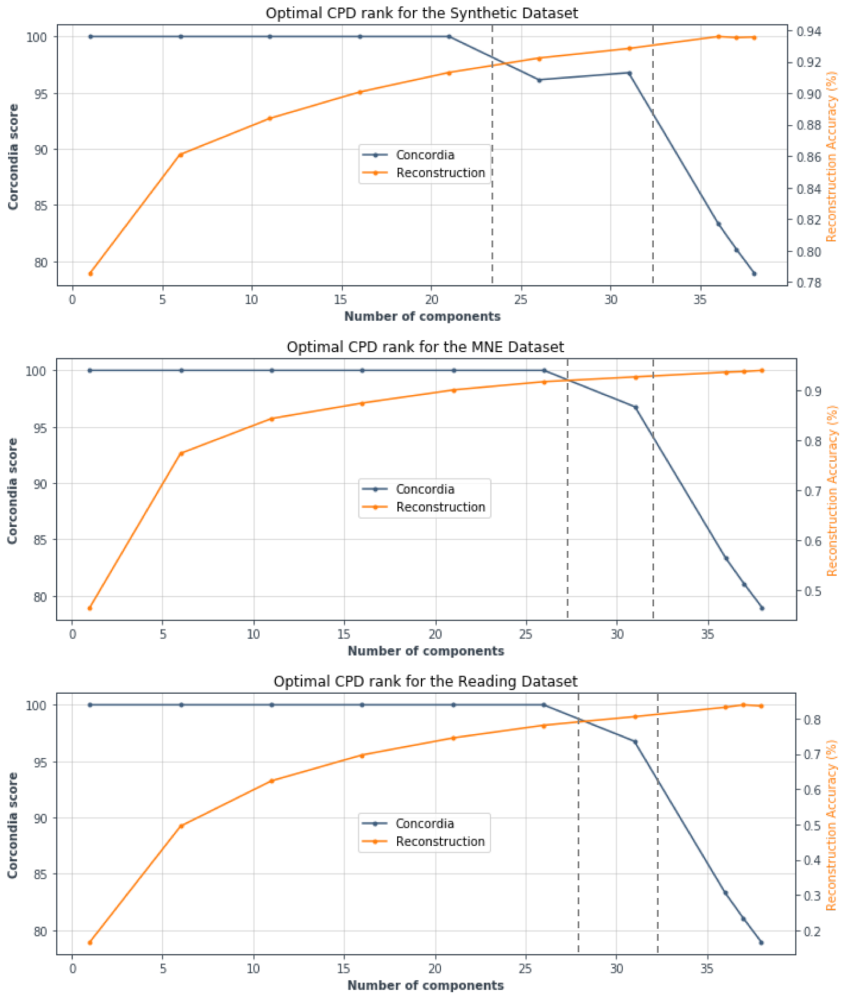

Figure 6.19: Corcondia score and the corresponding error for varying number of components in Parafac. The region highlighted between the vertical lines indicate the an appropriate rank for decomposing the dataset rank. (a) Synthetic dataset (b) Reconstructed dataset (c) Reading dataset

Randomised CPD

The addition of spectral information increases the size of the dataset by approximately 2 orders (×100 in other words). Due to this difference in the dataset size, even optimised tensor methods for efficiency are unlikely to achieve the same computational performance as blind source separation techniques based on matrix methods. Therefore an important point to consider would be computational complexity when comparing these algorithms. After all it is not ideal to have to segment the data, and thus lose some information for artifact removal. Three points of analysis were used to compare, the dimensionality, number of elements and rank when decomposing tensor dataset.

The theoretical complexity of the decomposition methods are calculated by com-

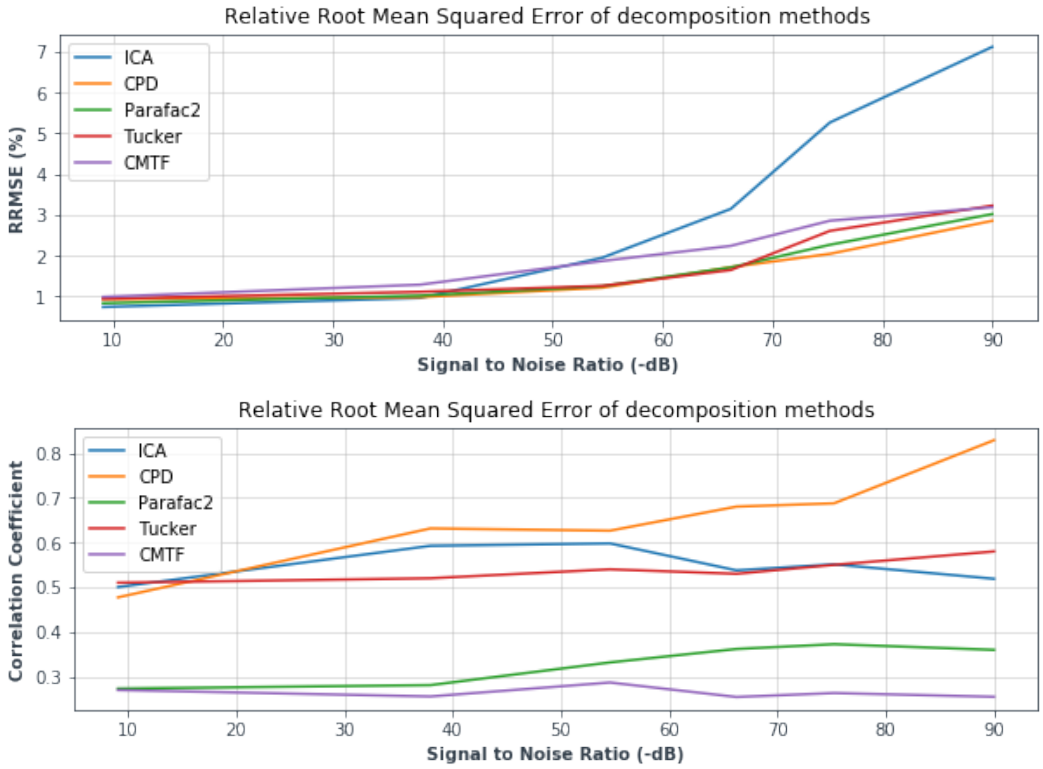

Figure 6.20: [Top] Relative Root Mean Squared Error (RRMSE) as on varying signal to noise ratios. [Bottom] Averaged Pearson's correlation coefficient on all electrodes over varying signal to noise ratios. Note: SNR values are very large to obtain a better distinction between the methods.

bining known complexities of the core operations performed in each iteration, as shown in Table 6.1. In the case of CPD, each iteration until convergence would have a time complexity of $\mathcal{O}(NR\prod_n I_n)$ (ignoring initialisation costs) [68]. A particular performance optimisation would be to sample uniformly with replacement, and use sampled Khatri-Rao to reduce the computational costs. This reduces the computational costs significantly to $\mathcal{O}(SR\sum_n I_n)$ [68]. The implementation listing is shown in 11.

Computing Process	Complexity
Gradient[69]	$\mathcal{O}(NRJ)$
Exact line search [69]	$\mathcal{O}(2^N RJ)$
Hessian and its inverse [69]	$\mathcal{O}(R^2T + NR^6)$
Kronecker product of all factors [68]	$\mathcal{O}(R^2[\sum_{m\neq n} I_m + N])$
Khatri-Rao / Sampled Khatri-Rao [68]	$\mathcal{O}(R[\prod_{m\neq n} I_m + N])$ / $\mathcal{O}(SR\sum_n I_n)$
Unfolding and multiplication [68]	$\mathcal{O}(2R\prod I_m)$

Table 6.1: Complexities of the sub-methods implemented for tensor methods. Given in terms of flops.

Algorithm 7: CPRAND using ALS

Data: Tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$, Rank R, Samples S
Result: $A^{(1)} \in \mathbb{R}^{I_1 \times R}, A^{(2)} \in \mathbb{R}^{I_2 \times R}, \ldots, A^{(N)} \in \mathbb{R}^{N \times R}, \boldsymbol{\lambda} \in \mathbb{R}^{1 \times R}$

Initialise factor matrices $\boldsymbol{A^{(n)}}$ **repeat**
  for $n \leftarrow k$ ***to*** N **do**
    $\boldsymbol{S} \leftarrow SamplingOperator \in \mathbb{R}^{S \times \prod_{m \neq n} I_m}$
    $\boldsymbol{V_S} \leftarrow SampledKhatriRao(\boldsymbol{S}, \boldsymbol{A}^{(1)T}, \ldots, \boldsymbol{A}^{(n-1)}, \boldsymbol{A}^{(n+1)}, \ldots, \boldsymbol{A}^{(N)}$
    $\boldsymbol{X}_S^{(\boldsymbol{T})} \leftarrow \boldsymbol{S}\boldsymbol{X}_{(n)}^T$
    $\boldsymbol{A}^{(n)} \leftarrow argmin_A \| \boldsymbol{V_S} bmA^T - \boldsymbol{X}_S^T \|_F$
    $\boldsymbol{\lambda} \leftarrow$ Norm of $\boldsymbol{A}^{(n)}$ columns
    Normalise $\boldsymbol{A}^{(n)}$ columns
  end
until *convergence criteria met*;
return $\boldsymbol{\lambda}, \boldsymbol{A}^{(N)}, \boldsymbol{A}^{(N-1)}, \ldots, \boldsymbol{A}^{(1)}$

The complexities were also calculated experimentally, to ensure their efficiencies match what is expected. Note the grey circles in the plots of figures 6.23, B.2 and 6.21 indicate that the algorithm had not converged.

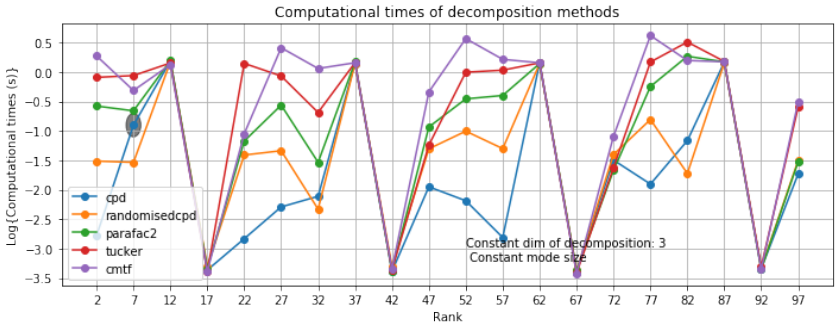

Figure 6.21: Computational times affected by the rank of the decomposition method, with constant number of elements and dimensions.

Discussion

RandomisedCPD did not improve upon the computational times to the significance expected. In many cases, CPRAND was slower for less computationally expensive tasks. One explanation for this could be that the implementation developed for this study used a particular access function (also developed during this project), unavoidably, which was repeatedly called. However a simple experiment suggests that the access function only scales linearly, which is better than the algorithms themselves, as shown in figure 6.24.

It is interesting to note a pattern in all graphs of Figure 6.23, B.2 and 6.23.

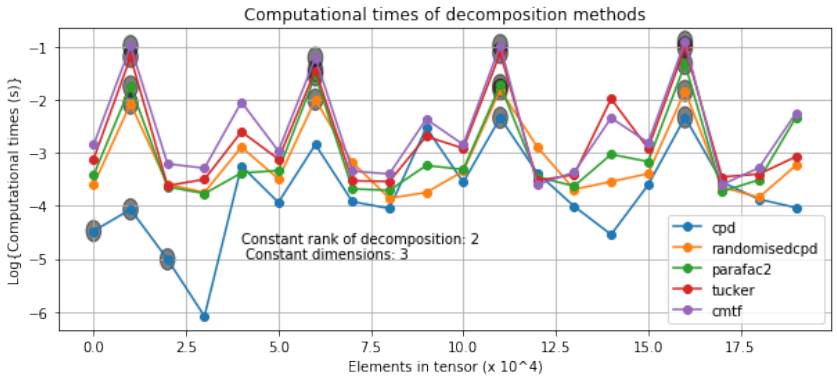

Figure 6.22: Computational times affected by the number of elements in the input, with constant dimensions and rank were used: 3 and 2 respectively.

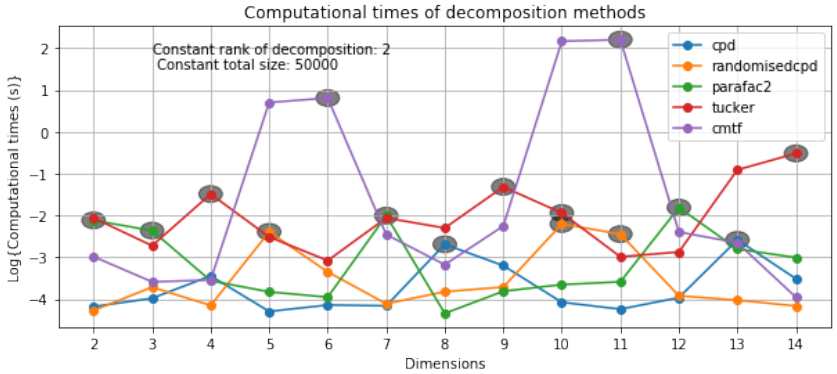

Figure 6.23: Computational times affected by the dimensionality of the tensor in the input, with constant number of elements and rank.

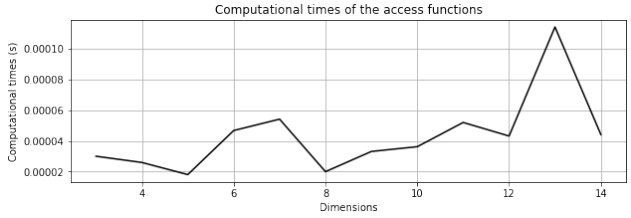

Figure 6.24: Computational times of the access function created. The access function is a big part of RandomisedCPD, and therefore a separate analysis is conducted .

Parafac2

PARAFAC2 uses the factors from the first iteration of PARAFAC. However there is a more subtle but efficient way of implementing this, without performing full PARAFAC. In-lining the decomposition showed significant performance improvements, by approximately 9.5%.

Tucker

Tucker is, in general, more computationally expensive than CPD or PARAFAC2. This is because performing tradition Singular Value Decomposition (SVD), as is required in this decomposition, using the QR algorithmis a very expensive operation. Briefly serving as a reminder, performing SVD of $\boldsymbol{A} = \Sigma \boldsymbol{V}^T$, an approximate of $\boldsymbol{A}$ is achieved. A general randomised technique for approximating $\boldsymbol{A}$, which is now also used for SVD, was developed by Halko et al. [70], and achieves considerable improvements to the performance.

7 — Applying decomposition for artifact removal on Real Data

A basic visual confirmation of the methods of quantification are shown in the appendix A.

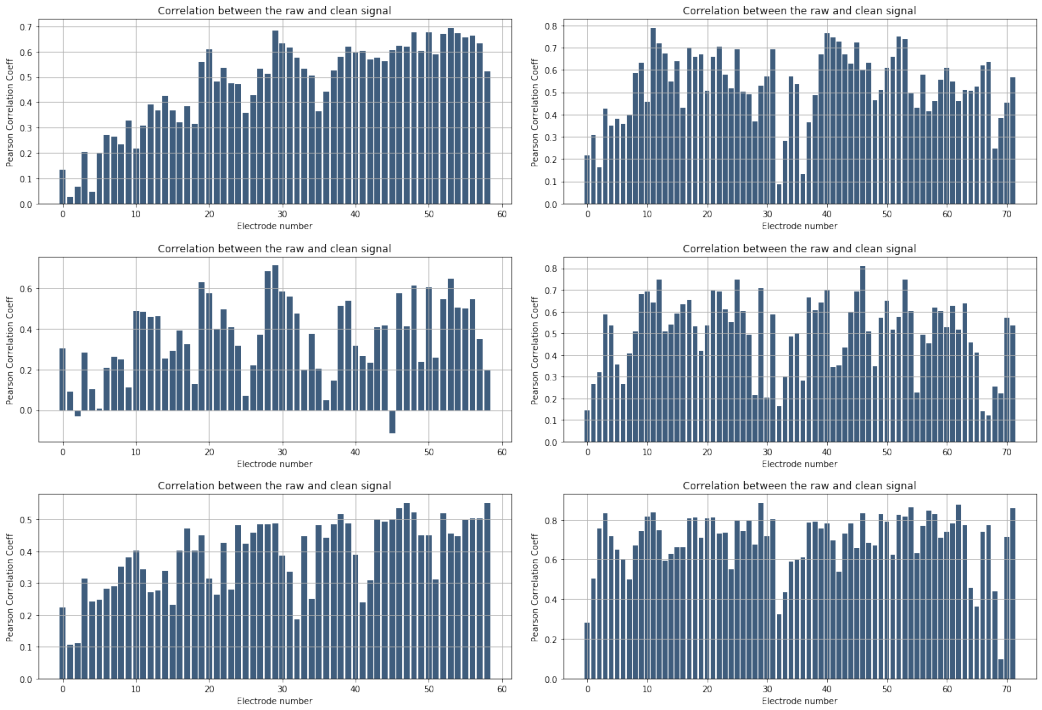

Figure 7.1: Pearson's correlation coefficients for all electrodes. The results of the MNE dataset are depicted on the left, whilst the right has the reading dataset. (a) Parafac (b) Tucker (c) Parafac2

Qualitatively, a clear distinction in the information extracted in the components of each decomposition methods is also visible. For the purpose of brevity, this has been kept in appendix B. Parafac and Parafac2 are able to better discriminate. From Chapter 5, an algorithm that has performed well will have low values of correlation that are gradually increasing as the artifact power is reduced. In the case of the MNE dataset, the best performance was achieved with Parafac Decompo-

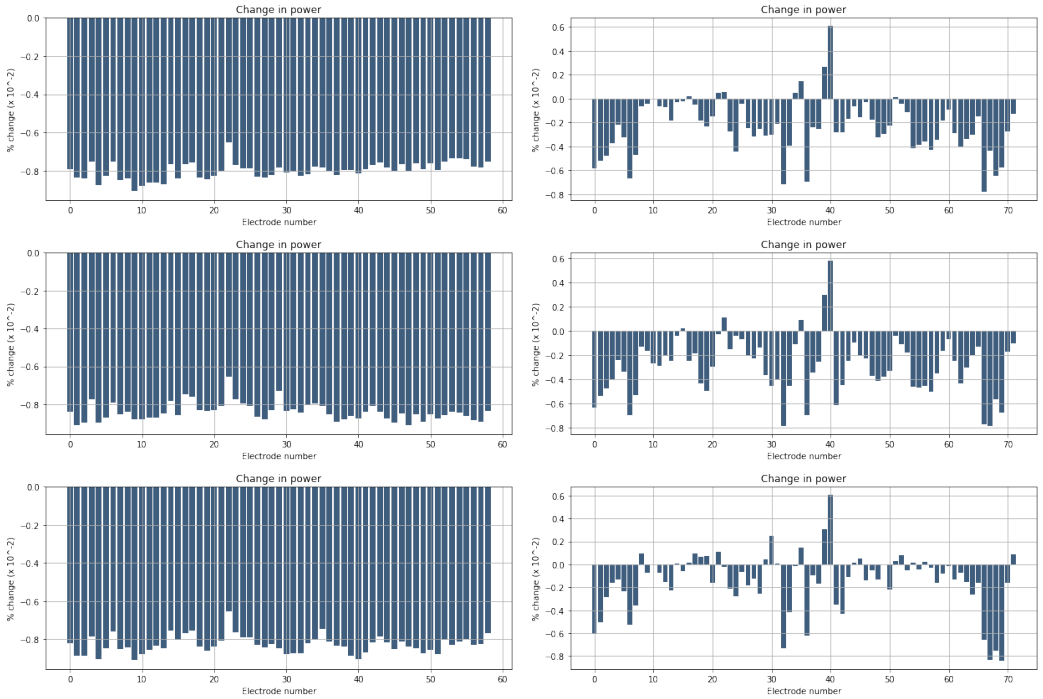

Figure 7.2: Percentage change in power for all electrodes. The results of the MNE dataset are depicted on the left, whilst the right has the reading dataset. (a) Parafac (b) Tucker (c) Parafac2

sition. For the *Reading* dataset, determining a best performing algorithm is less straightforward. Tucker and CPD both exhibit a gradual change, whilst a more sudden change was observed for the Parafac2 decomposition. Similar correlation breaks were observed in Tucker for particular electrodes, suggesting that a certain proportion of neural activity was also extracted from the signal. Interestingly all three decomposition strategies have low correlations for electrodes that are spatially placed in the middle of the scalp. This could be due to the difficulty of isolating a component associated with blinks if the neural activity from the mid region of the brain is severely contaminated in the noisy EEG signal.

Similarly the largest percentage decrease in power should be for occipital channels. In the case of the *Reading* dataset, all methods removed a significant proportion of the signal's power on the back of the scalp. This is a strong indication that the removed components were not merely blink artifacts. Therefore unless there was contamination from the EOG channel to the electrodes in this location of the scalp, the method was less successful. for this dataset. Fewer electrodes had a significant power change in Parafac2 relative to the other methods. From the wider analysis (not all has been included in the thesis), it is difficult to determine which tensor method performed the best. It is important to note that, without ground truth, these metrics cannot form as a method of ensuring that the decomposition methods have removed blinks successfully. They are used as an estimate to performance,

but are not measures of performance themselves.

Reconstruction errors were also considered. It was expected that Parafac2 have the smallest reconstruction error due to its superior general representation. Similarly CMTF performs ALS to minimise errors on a tensor and a matrix. The observations match predictions exactly in this case, as shown in Figure 7.3.

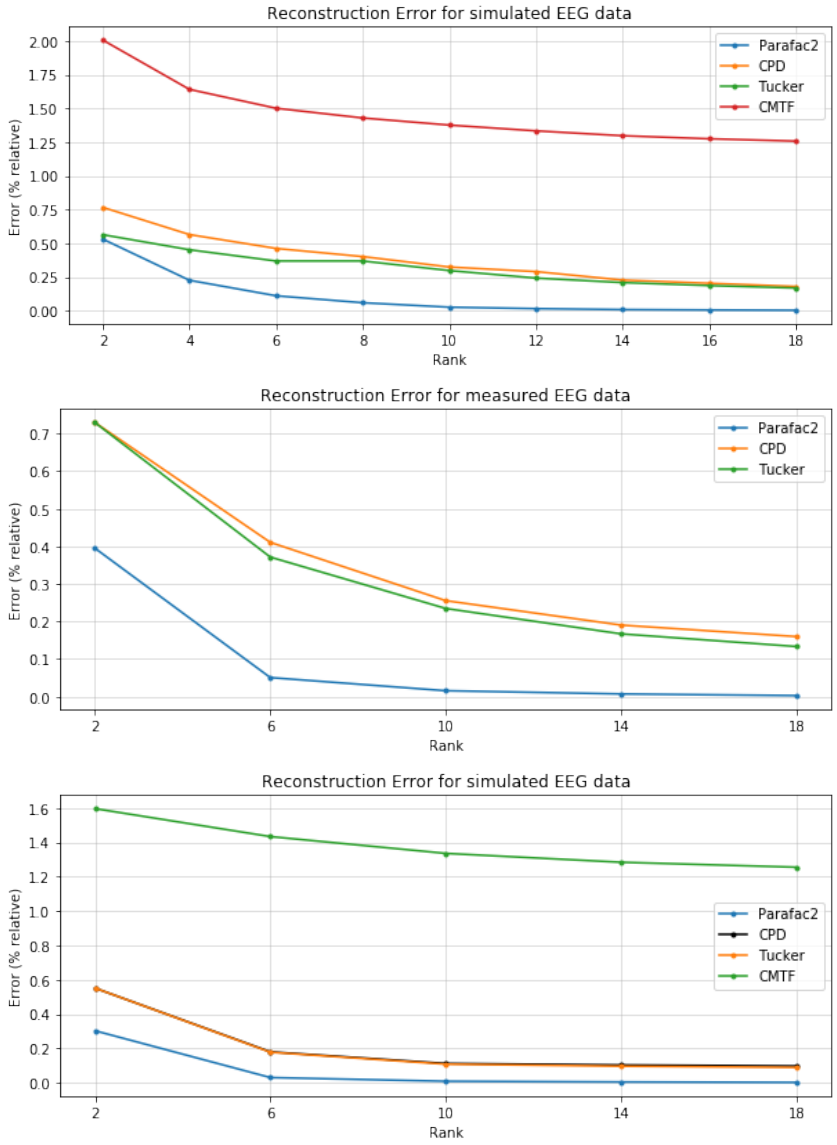

Figure 7.3: Reconstruction error over a short time duration (≈ 3 seconds giving 10 million elements). (a) Synthetic dataset (b) Real MNE dataset (c) Real Reading dataset

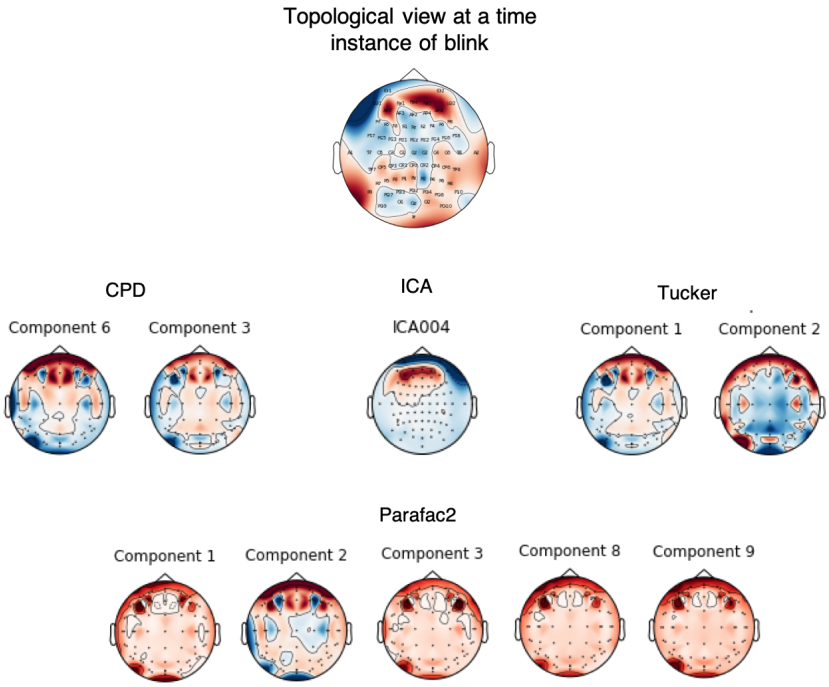

Figure 7.4: An example of the topological view of the blink in the Reading dataset. The components with EOG artifact information from CPD, ICA, Tucker and Parafac2 are illustrated in their spatial representation.

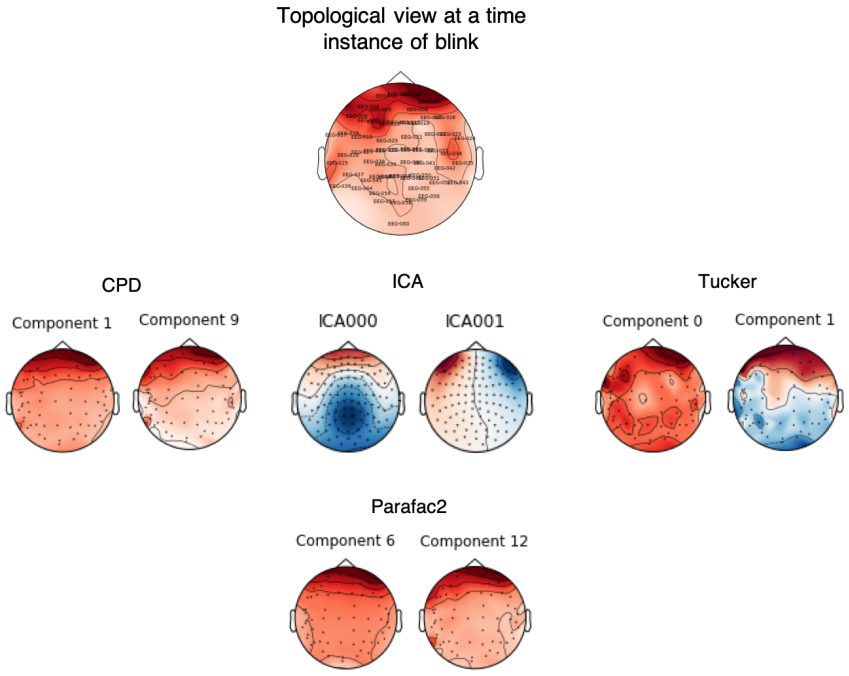

Figure 7.5: An example of the topological view of the blink in the MNE dataset. The components with EOG artifact information from CPD, ICA, Tucker and Parafac2 are illustrated in their spatial representation.

8 — Tensor Toolbox

This chapter is not intended as a user guide to HottBox, as that was not the sole purpose of the project. A separate tutorial has, however, been developed for the implementations in this study should the reader wish to explore the library. It can be found in [71].

8.1 Introduction

Hottbox [3] is an open source library developed at Imperial College London with the purpose of providing tensor methods. A collaboration between Department of Computing at Imperial and Caltech has produced a similar library in Python [**?**]. However it has a larger focus on the computational aspects of decomposition and producing efficient algorithms. Therefore it is different to HottBox in the sense that it does not address all aspects from visualisation to feature extraction. Another library in Matlab [1] has the same objective, however the language is somewhat less popular in industry due to its lack of ease in compatibility with Hadoop and Spark for example.

Previous work

The library was initially developed by Ilya Kisil as part of his PhD Thesis. Prior to this project, it had three decomposition methods from tensor literature implemented: CPD, Tucker, Tensor Train. A number of useful tools such as the integration with Pandas, meta-data considerations and careful software design was also already embedded into the library.

8.2 Development

Implementations

The following decomposition methods were added: Randomised CPD and Parafac2 and some modifications to the current ones were also made. CMTF for data fusion

[1] https://www.tensorlab.net/

was implemented, along with many software engineering refactoring and additions. All tensor methods mentioned in this report were implemented on a Jupyter Notebook and thereafter embedded into the library.

Other significant implementations

Synthetic Tensors

Structured tensors such as Toeplitz tensors were defined as multilinear generalisations of their matrices. Though they have not been studied in tensor literature, the Toeplitz/Hankel structures have desirable properties in the application of signal processing. It has been exploited in tensor networks by Cichocki et al. [14]. It is possible to represent several signal processing problems as tensor decompositions with Toeplitz structured factor matrices. In relation to this project, the Toeplitz structure can be exploited in blind source separation techniques using block tensor decompositions. The Toeplitz tensor generation has been implemented in the library, but it is left as an open problem to apply it to EEG signals. The implementation required the addition of an access function to be able generalise the implementation to an N-way tensor. Example applications in signal processing [72] and blind source separation [73].

One interesting property of the structured Toeplitz tensors is that it represents n-order moments for a tensor of dimension n. Consider the third dimensional case shown in equation 8.1, it is shown in [74] that this is a symmetric tensors, and in particular a Toeplitz tensor.

A Toeplitz tensor χ is defined such that for all permutation π any sub-matrix of the tensor $\boldsymbol{M}_k^{(\pi)}$where $k \in [0...I_{\pi_3} - 1]$ is a Toeplitz matrix. The simple proof of which is given in [74]. This definition works well with tensors of equal modes, but breaks down at other mode sizes. Therefore for Hottbox's implementation, modes are required to be specified along which it is expected that Toeplitz matrices are formed. A naive approach to the problem was employed, where all slices in each mode are iteratively set to a Toeplitz matrix (randomly generated if not user input). No alternative implementation of the generation of this structured tensor is available (to the best of the author's knowledge).

As was expected the algorithms were able to converge significantly faster with structured tensors. An example case for CPD is shown in figure 8.1.

$$[\mathcal{M}_{t_1,t_2,t_3}] = E\left(x(t_1)x(t_2)x(t_3)\right) \tag{8.1}$$

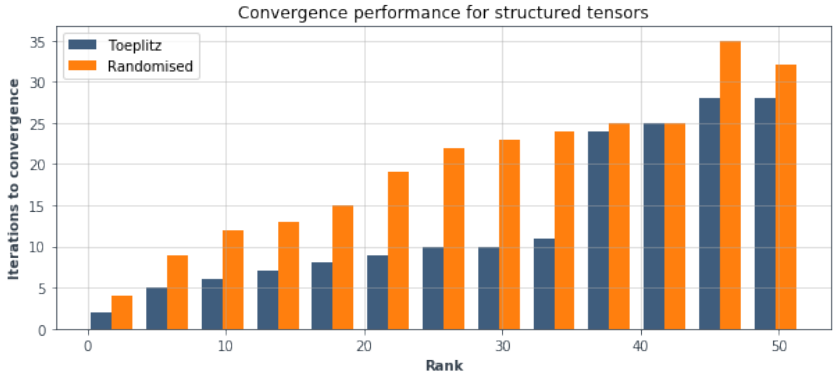

Figure 8.1: A randomly generated tensor and a Toeplitz tensor of the same size ($50 \times 50 \times 50$) are tested for iterations to convergence for a number of ranks. CPD is able to converge on a Toeplitz tensor in less iterations for all tests.

A general class of synthetic generation was also added that enabled the user to specify the distribution, modes and dimensions, as well as whether the tensor should be sparse or dense. For example in the context of EEG noise, each channel can be modelled to have an independent Gaussian noise, if the Gaussian distribution over slices is specified. This allows for quick, naive modelling in a variety of contexts.

Testing

The implemented algorithms were principally tested by replicating their unique properties claimed in literature. In the case of CPD, each component should have a discernible physical meaning - which can be seen in Figure 8.2 as an example. A similar sanity check was performed for testing the Parafac2 and CPDRand implementation. In the case of Tucker, it is known that the matrices are placed in order of decreasing Frobenius-norm. Therefore the n components obtained from a rank of n, should be the same as the first n from rank $n+1$. This is demonstrated in Figure8.3.

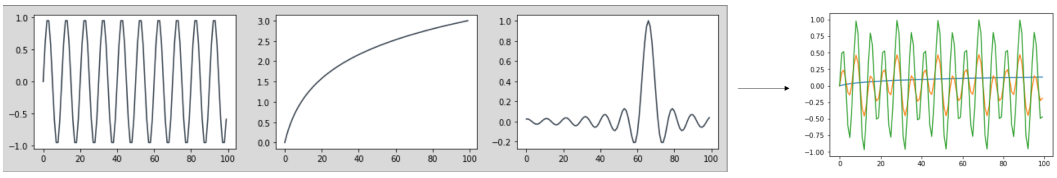

Figure 8.2: Three types of signal of varying frequency are placed in fibers of different modes of the tensor, with the components extracted from CPD of rank three shown on the right. CPD was able to separate the sinusoidal, log and sinc function provided.

Note: there is no physical interpretation to the random signals of Figure 8.2 and 8.3, therefore axis have been chosen to be omitted.

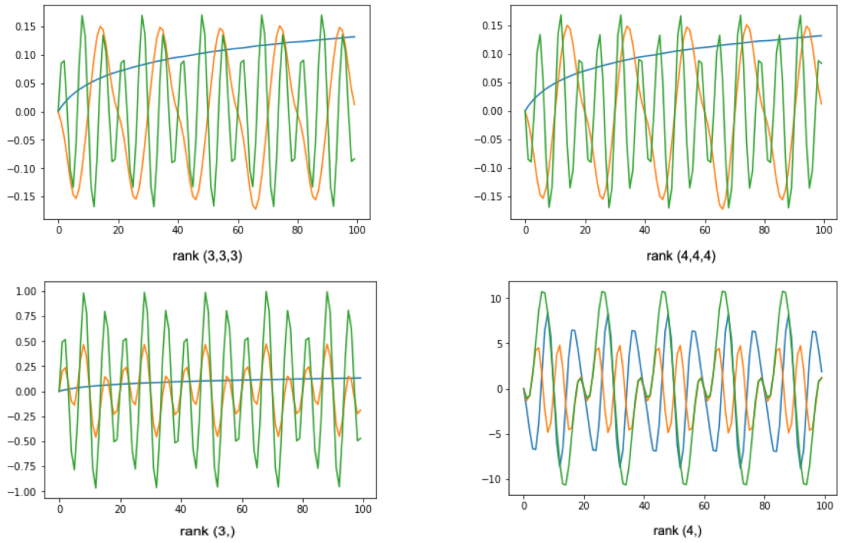

Figure 8.3: (a) Tucker (HOSVD) Decomposition with the first 3 components selected from $rank(3,3,3)$ *and* $rand(4,4,4)$*. (b) Parafac Decomposition with the first 3 components selected from* $rank3$ *and* $rand3$*. It is clearly observed that the first n components are the same in the case of Tucker for varying ranks, but different in the case of CPD.*

The implementations were also tested using unit tests developed for each algorithm. In Python a popular library '*pytest*' has been used for writing these tests. Sufficient measures were taken to ensure good functional programs were written for the implementations. For example, the decomposition methods had base classes, forcing decomposition, initialisation and reconstruction to be split for any current and future use. This ensures the functions are well exposed for testing, and each component of the implementation can be tested independently of the others. There are many useful practices for unit testing, which will not be discussed in this thesis - but many of which have been implemented in HottBox.

9 — Strengths and Limitations of Tensor Decompositions for EEG

This was discussed theoretically in chapter 4, and will now be tackled from a practical perspective. One simple manner of comparison is to view the extracted components from each dataset with the optimal rank (determined by Core Consistency Analysis). It was hypothesised that Tensor methods will be superior in their extraction of the blink artifacts. It has been illustrated, though not yet explicitly stated, that this was the case for the synthetic data shown through Figures 6.18, 6.6, 6.3. However consider now the real signals: Reading and MNE. Using all datasets, the extracted components from the tested decomposition methods are shown in Figures 7.4 and 7.5. In the case of the MNE dataset, Parafac and Parafac2 outperformed ICA and Tucker in localising the artifact components. Both ICA and Tucker weakly represent the EOG artifacts, and the components also present other captured activity. Similarly Tucker and CPD are observed to have the noisier *Reading* dataset. Corroborating the claimed superior localisation abilities of tensor methods further, the result of the synthetic dataset (refer to Figure **??**) have demonstrated better artifact removal with high SNRs.

9.1 Removing Line noise

The line noise was not filtered out in the reading dataset, allowing further exploration of the tensor methods. It is common practice to use a notch filter at 50 Hz (or 60 Hz, depending on the A/C mains frequency) to remove a large noise component. This is easy to carry out because the artifact can be considered as *'well-behaved'* noise, following an almost Gaussian distribution. Due to this there have not been many other research efforts in isolating line noise artifacts. However any activity at this frequency will be discarded as a result of filtering. Although there is unlikely to be any neural activity at this frequency, it may be that other activities at those frequencies recorded are useful. Instead, a similar approach to removing EOG artifacts has been applied in this study to remove line noise, as depicted in

figure 9.1.A Jarque-Bera test may be appropriate for determining whether the data is normally distributed, and therefore which components to remove. However this is only recommended for values larger that ≈ 2000, which would require the number of electrodes to be of that value. Therefore such tests are not feasible in EEG analysis, and a manual qualitative assessment is used.

A matrix method such as ICA can be used theoretically, however due to the missing spectral information it is highly unlikely that line noise power will be described by one or two components.

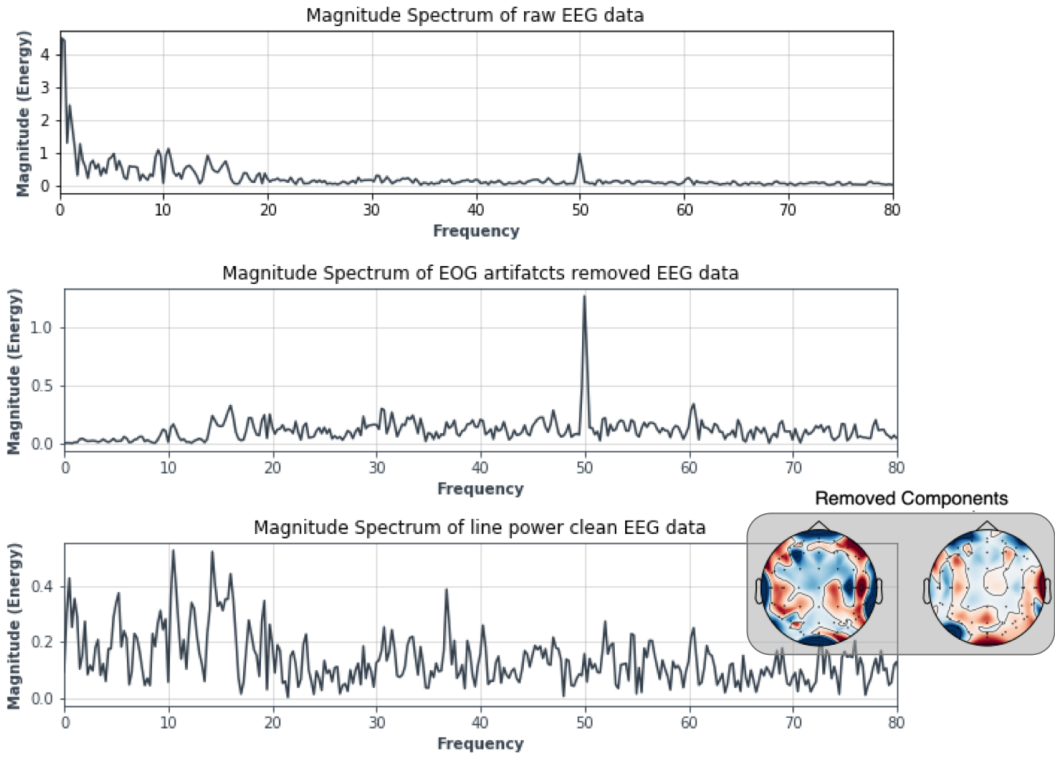

Figure 9.1: Magnitude spectrum plots depicting line artifact removal using Tucker decomposition of rank 20. The raw dataset is cleaned (only for visual purposes) of EOG artifacts to emphasise the spectrum of the raw dataset. A spike is clearly observed at 50 Hz. All the power related to this noise were explained by two components, removing which a cleaner signal was obtained. The power (measured by RMS) was reduced from 103.5 to 7.93 in relative units.

9.2 Information retrieval on corrupted data

Now low pass filtering + line filtering + harmonics at irregularly spaced frequencies. ("ICA revealed presence of alpha activity near 10 Hz which was highly obscured in original data").

9.3 Higher dimensions

The datasets, synthetic and real, used in this study adequately additional information for an adequate isolation of artifacts. One advantage of using tensor decompositions is that additional information for better localisation can always be achieved, which involves forming a higher dimensional tensor. s was mentioned in section 3.6, more channels may be used such as trials and subjects to allow for a better representation of these components. This will only occurs if similar tasks are carried out during the recording, with the desired activity prevalent in all tasks. In the datasets tested, it was not found that additional information regarding the trials or the subjects are able to shape into better artifact removal methods. Therefore not more than three dimensions, corresponding to channels, time and frequency, were used.

9.4 Limited data

The tensor methods or matrix methods relevant to this study are fundamentally statistical multi-linear analysis of the datasets. They do not predict, nor make any other abstractions from the seen data. Therefore the results obtained from these analysis are only meaningful if a sufficiently large dataset is used. Practical observations have suggested that the amount of variation present in the dataset seems to be more important than the number of points in the dataset, which would be expected. This is true for any statistical approach, but is particularly important for EEG signals as they are noisy measurements during a certain task (which may be resting or reading). If the task is to respond to a stimulus on the screen, it is unlikely that the EOG artifacts will be present during the short burst of neural activity during this time for example.

Efficiency is an important issue with tensor methods in this application. This is because transforming time components into spectral components using CWT or FFT of size n, will multiply the size of the dataset by n. It would therefore not be feasible to conduct this research on very large datasets. Though efficiency has been considered in this study, and *RandCPD* implemented, it is unlikely that more than 15 seconds of EEG data of a typical frequency (512 Hz) can be computed in reasonable time with easily accessible resources.

10 — Evaluation and Discussion

The work to this end exploits the superior properties of tensor decompositions, such as CPD, over matrix methods, in this case ICA, to better localise activity in EEG signals. In the processing of such signals, a better localisation would not only be able to remove more noise (albeit biological artifacts or instrumental noise) but also minimise the neural activity removed. Maintaining important cerebral activities that may provide vital biological insights is a very important factor that will enable the noisy, non-invasive technique to become more interpretable. Most studies state their results through an illustration in either the spatial, temporal or spectral domain rather than quantifying the findings. As in the thesis in [75], different tensor methods were evaluated and compared in this project. Although Parafac2 was explored as in for this project, data fusion and efficiency were not and a much larger focus of synthetic data generation was placed.

The tensor methods to localise activity were evaluated using three datasets: synthetic (ideal), well behaved filtered measured and a raw noisy signal. A quantification method for analysing the results was defined in chapter 5, in both the cases of an available ground truth and one not. A quantification method considering the desirable outcomes to neurologists, properties of continuous wavelet transforms, and the reasons for artifacts to occur was established. The approach is not necessarily a unique one or unified (into a single metric) - but nonetheless tested. It had previously been speculated that a unified quantification method using probabilistic signal processing could be developed, however the uncertainty with the method was too large.

The software developed is placed in an open source library, allowing any reader to replicate the results found in this project. Certain implementations are not found in any other library readily available online, such as that of PARAFAC2, and it is therefore hoped that the library will encourage more scientific contributions in the field of tensors. Although unit-testing and care was taken in developing the algorithms, the implementations themselves were evaluated entirely based on ideas and intuitions gathered through the properties of the tensor method described in

literature. This was described in more detail in Section 8.2. However no method of unifying the testing of the decomposition methods introduced in chapter 4 and chapter 3 were developed. There is, therefore, a lack of foundations for testing implementations - which forms as possibilities of further work. By allowing software to be open source, it is also hoped that the wider community accessing such resources will contribute in bettering them.

Theoretical justification of decomposing EEG signals with tensors was provided in Chapter 4, building on known properties of tensor methods seen in previous literature. It was found, as seen in figures B.2, that tensor decompositions are able to localise activity better in cases of higher noise. The example of identifying and removing line noise power using tensor decompositions is novel, and has demonstrated that tensors enable insights that other matrix methods cannot readily provide.A mathematical reasoning of one of the methods of blind source separation (for the removal of an activity) in tensors from the perspective of linear algebra was also provided in the work, which seemed to be lacking in previous literatures. Using this it was possible to compare the performance of all the considered tensor decomposition strategies.

Visually the components extracted, should a human manually examine them, for CPD were seen to localise activity better. It remains an open question whether this is practically useful, due to the significantly increased computational resources required, when performing simultaneous time-frequency analysis. On the lowest SNR value, ICA had a 12.4% smaller RRMSE than the best tensor method (Parafac2). On the highest SNR value, the worst performing tensor method (Tucker) had a 55 % lower RRMSE than ICA. Of the explored tensor decomposition methods, CMTF was consistently the worst performing. This may be because EEG had been fused with MEG, both of which have high temporal resolutions and measure electrical activity directly or indirectly. A different choice would have been to use fMRI and EEG as they have complementary weaknesses: the lack of temporal specificity of an activity and the lack of spatial specificity respectively. Numerous studies have shown improved localisation of activity when using both datasets[76]. This remains an open question in this study.

11 — Conclusion and Further work

Blind Source Separation (BSS) based on tensor decomposition methods have demonstrated to have more desirable properties that are able to exploit brain signals better, as was theoretically hypothesised. Due to the easy natural abstraction from the well developed and studied ideas of BSS using Independent Components Analysis (ICA) to its multi-linear form of tensors, this matrix method was used as the baseline comparison. In the case of an excellent signal to noise ratio, no tensor decomposition was able to outperform ICA. However at worse SNRs tensors were able to maintain their localisation performance better due to the additional information from Continous Wavelet Transforms. Synthetic dataset was created for this analysis.

Through this analysis tensors are suggested to be the statistical framework of choice for EEG analysis, as they enable better interpretability as well as serving as a single and unified tool for many kinds of analysis. The framework they provide is suited to many purposes, and those demonstrated in this work include simultaneous time-frequency analysis, the removal of unwanted components such as line noise power and artifact removal. There was little difference in the performance obtained between the tensor methods for unsupervised BSS of artifacts, however Parafac and Parafac2 were found to be better than Tucker for the datasets that had been employed. Parafac2 did not experimentally outperform CPD, by any significant measure, as had been anticipated.

11.1 Further work

The results stated throughout this project were through the analysis of space-time-frequency of EEG signals. However, as was briefly discussed in section 9.3, it is possible to incorporate more modes for a better insight into the particular dataset. An open question that remains for example is whether having more subjects conducting the same activity enables better localisation of artifacts. This would require greater resources, to be able to experimentally gather them as well as more compute resources. The complexity of the implementation is often seen to be a

major drawback, and further optimisations of sparse datasets are possible in the future.

A probabilistic approach using Bayesian statistics is traditionally used to incorporate a priori information, making it a semi-blind source separation problem. One method would be to incorporate physiological constrains to cost functions, and using gradient based minimisations. This class of algorithms are referred to as Constrained Blind Source Separation. It was left to future work due to the limited time available to be able to understand the particular dataset better. Development for specific artifacts have been made, such as blinks [77]. In addition to the Core Consistency Analysis used, Bayesian Information Criteria (BIC) could have also been used as an optimisation for the number of components.

A further discussion could compare the representation of Parafac2 and time-drift corrected real EEG (instead of the raw EEG currently used) using Parafac in greater depth. Possible methods of correction include subtraction by minimum phase subtraction [78]. Similarly only fastICA has been used, without any exploration into the better matrix methods due to time constraints. Datasets of more meaningful activity, such as epileptic seizures, could also posses richer properties that tensors of higher dimensions are able to extract.

Bibliography

[1] Mario Tudor, Lorainne Tudor Car, and Katarina Ivana Tudor. Hans berger (1873-1941) - the history of electroencephalography. *Acta medica Croatica : casopis Hravatske akademije medicinskih znanosti*, 59:307–13, 02 2005. pages

[2] J. Mocks. Topographic components model for event-related potentials and some biophysical considerations. *IEEE Transactions on Biomedical Engineering*, 35(6):482–484, June 1988. ISSN 0018-9294. doi: 10.1109/10.2119. pages

[3] I Kisil, A Moniri, G G Calvi, B Scalzo Dees, D Manocha, and D P Mandic. Hottbox: Higher order tensors toolbox, 2017. URL `https://github.com/hottbox/hottbox`. pages

[4] Rasmus Bro. Parafac. tutorial and applications. *Chemometrics and Intelligent Laboratory Systems*, 38(2):149171, 1997. doi: 10.1016/s0169-7439(97)00032-4. pages

[5] Applications. *Multi-Way Analysis with Applications in the Chemical Sciences*, page 257349, Mar 2005. doi: 10.1002/0470012110.ch10. pages

[6] J. Douglas Carroll and Jih-Jie Chang. Analysis of individual differences in multidimensional scaling via an n-way generalization of eckart-young decomposition. *Psychometrika*, 35(3):283319, 1970. doi: 10.1007/bf02310791. pages

[7] Henk A. L. Kiers and Iven Van Mechelen. Three-way component analysis: Principles and illustrative application. *Psychological Methods*, 6(1):84110, 2001. doi: 10.1037/1082-989x.6.1.84. pages

[8] Ledyard R Tucker. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3):279311, 1966. doi: 10.1007/bf02289464. pages

[9] Andrzej Cichocki, Danilo P. Mandic, Anh Huy Phan, Cesar F. Caiafa, Guoxu Zhou, Qibin Zhao, and Lieven De Lathauwer. Tensor decompositions for signal processing applications: From two-way to multiway component analysis. *IEEE Signal Processing Magazine*, 32:145–163, 2015. pages

[10] M. A. O. Vasilescu and Demetri Terzopoulos. Multilinear analysis of image ensembles: Tensorfaces. In *Proceedings of the 7th European Conference on Computer Vision-Part I*, ECCV '02, pages 447–460, London, UK, UK, 2002. Springer-Verlag. ISBN 3-540-43745-2. URL http://dl.acm.org/citation.cfm?id=645315.649173. pages

[11] Evrim Acar, Daniel M. Dunlavy, Tamara G. Kolda, and Morten Mrup. *Scalable Tensor Factorizations with Missing Data*, pages 701–712. doi: 10.1137/1.9781611972801.61. URL https://epubs.siam.org/doi/abs/10.1137/1.9781611972801.61. pages

[12] T. Kolda and B. Bader. Tensor decompositions and applications. *SIAM Review*, 51(3):455–500, 2009. doi: 10.1137/07070111X. URL https://doi.org/10.1137/07070111X. pages

[13] Andrzej Cichocki, Namgil Lee, Ivan Oseledets, Anh-Huy Phan, Qibin Zhao, and Danilo P. Mandic. Tensor networks for dimensionality reduction and large-scale optimization: Part 1 low-rank tensor decompositions. *Foundations and Trends in Machine Learning*, 9(4-5):249429, 2016. doi: 10.1561/2200000059. pages

[14] A. Cichocki, A-H. Phan, Q. Zhao, N. Lee, I. V. Oseledets, M. Sugiyama, and D. Mandic. Tensor Networks for Dimensionality Reduction and Large-Scale Optimizations. Part 2 Applications and Future Perspectives. *arXiv e-prints*, art. arXiv:1708.09165, Aug 2017. pages

[15] Evangelos E. Papalexakis, Christos Faloutsos, and Nicholas D. Sidiropoulos. Tensors for data mining and data fusion: Models, applications, and scalable algorithms. *ACM Trans. Intell. Syst. Technol.*, 8(2):16:1–16:44, October 2016. ISSN 2157-6904. doi: 10.1145/2915921. URL http://doi.acm.org/10.1145/2915921. pages

[16] Lars Grasedyck, Daniel Kressner, and Christine Tobler. A literature survey of low-rank tensor approximation techniques. 2013. pages

[17] J. M. Papy, L. De Lathauwer, and S. Van Huffel. Exponential data fitting using multilinear algebra: the single-channel and multi-channel case. *Numerical Linear Algebra with Applications*, 12(8):809–826, 2005. doi: 10.1002/nla.453. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/nla.453. pages

[18] Boris N. Khoromskij. Fast and accurate tensor approximation of a multivariate convolution with linear scaling in dimension. *Journal of Compu-*

tational and Applied Mathematics, 234(11):3122 – 3139, 2010. ISSN 0377-0427. doi: https://doi.org/10.1016/j.cam.2010.02.004. URL http://www.sciencedirect.com/science/article/pii/S0377042710000750. Numerical Linear Algebra, Internet and Large Scale Applications. pages

[19] Richard A. Harshman, Peter Ladefoged, H. Graf von Reichenbach, Robert I. Jennrich, Dale Terbeek, Lee Cooper, Andrew L. Comrey, Peter M. Bentler, Jeanne Yamane, and Diane Vaughan. Foundations of the parafac procedure: Models and conditions for an "explanatory" multimodal factor analysis. 2001. pages

[20] N. D. Sidiropoulos and R. Bro. On the uniqueness of multilinear decomposition of n-way arrays. *Journal of Chemometrics*, 14:229–239, 2000. ISSN 0886-9383. pages

[21] Joon Hee Choi and S. V. N. Vishwanathan. Dfacto: Distributed factorization of tensors. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'14, pages 1296–1304, Cambridge, MA, USA, 2014. MIT Press. URL http://dl.acm.org/citation.cfm?id=2968826.2968971. pages

[22] Fumikazu Miwakeichi, Eduardo Martnez-Montes, Pedro A. Valds-Sosa, Nobuaki Nishiyama, Hiroaki Mizuhara, and Yoko Yamaguchi. Decomposing eeg data into spacetimefrequency components using parallel factor analysis. *NeuroImage*, 22(3):1035 – 1045, 2004. ISSN 1053-8119. doi: https://doi.org/10.1016/j.neuroimage.2004.03.039. URL http://www.sciencedirect.com/science/article/pii/S1053811904001958. pages

[23] F Estienne, N Matthijs, D.L Massart, P Ricoux, and D Leibovici. Multi-way modelling of high-dimensionality electroencephalographic data. *Chemometrics and Intelligent Laboratory Systems*, 58(1):59 – 72, 2001. ISSN 0169-7439. doi: https://doi.org/10.1016/S0169-7439(01)00140-X. URL http://www.sciencedirect.com/science/article/pii/S016974390100140X. pages

[24] Morten Mrup. Applications of tensor (multiway array) factorizations and decompositions in data mining. *Wiley Interdisc. Rew.: Data Mining and Knowledge Discovery*, 1:24–40, 01 2011. doi: 10.1002/widm.1. pages

[25] Evrim Acar, Rasmus Bro, and Age K. Smilde. Data fusion in metabolomics using coupled matrix and tensor factorizations. *Proceedings of the IEEE*, 103: 1602, 09 2015. doi: 10.1109/JPROC.2015.2438719. pages

[26] L. Eldn and B. Savas. A newtongrassmann method for computing the best multilinear rank-$(r_1,$ $r_2,$ $r_3)$ approximation of a tensor. *SIAM Journal on Matrix Analysis and Applications*, 31(2):248–271, 2009. doi: 10.1137/070688316. URL https://doi.org/10.1137/070688316. pages

[27] Nick Vannieuwenhoven, Raf Vandebril, and Karl Meerbergen. A new truncation strategy for the higher-order singular value decomposition. *SIAM Journal on Scientific Computing*, 34:1027–1052, 04 2012. doi: 10.1137/110836067. pages

[28] Ning Liu, Benyu Zhang, Jun Yan, Zheng Chen, Wenyin Liu, Fengshan Bai, and Leefeng Chien. Text representation: from vector to tensor. In *Fifth IEEE International Conference on Data Mining (ICDM'05)*, pages 4 pp.–, Nov 2005. doi: 10.1109/ICDM.2005.144. pages

[29] J. G. Nagy and M. E. Kilmer. Kronecker product approximation for preconditioning in three-dimensional imaging applications. *IEEE Transactions on Image Processing*, 15(3):604–613, March 2006. ISSN 1057-7149. doi: 10.1109/TIP.2005.863112. pages

[30] Berkant Savas and Lars Eldn. Handwritten digit classification using higher order singular value decomposition. *Pattern Recognition*, 40(3):993 – 1003, 2007. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2006.08.004. URL http://www.sciencedirect.com/science/article/pii/S0031320306003542. pages

[31] Lieven De Lathauwer and Joos Vandewalle. Dimensionality reduction in higher-order signal processing and rank-(r1,r2,,rn) reduction in multilinear algebra. *Linear Algebra and its Applications*, 391:31 – 55, 2004. ISSN 0024-3795. doi: https://doi.org/10.1016/j.laa.2004.01.016. URL http://www.sciencedirect.com/science/article/pii/S0024379504000886. Special Issue on Linear Algebra in Signal and Image Processing. pages

[32] I. Oseledets. Tensor-train decomposition. *SIAM Journal on Scientific Computing*, 33(5):2295–2317, 2011. doi: 10.1137/090752286. URL https://doi.org/10.1137/090752286. pages

[33] David Perez-Garcia, Frank Verstraete, Michael M Wolf, and J Ignacio Cirac. Matrix product state representations. *arXiv preprint quant-ph/0608197*, 2006. pages

[34] Ulrich Schollwöck. The density-matrix renormalization group. *Reviews of modern physics*, 77(1):259, 2005. pages

[35] A Novikov, Anton Rodomanov, Anton Osokin, and Dmitry Vetrov. Putting mrfs on a tensor train. *31st International Conference on Machine Learning, ICML 2014*, 3:2388–2399, 01 2014. pages

[36] Alexander Novikov, Dmitry Podoprikhin, Anton Osokin, and Dmitry Vetrov. Tensorizing neural networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, pages 442–450, Cambridge, MA, USA, 2015. MIT Press. URL http://dl.acm.org/citation.cfm?id=2969239.2969289. pages

[37] L. De Lathauwer, B. De Moor, and J. Vandewalle. On the best rank-1 and rank-(r1 ,r2 ,. . .,rn) approximation of higher-order tensors. *SIAM Journal on Matrix Analysis and Applications*, 21(4):1324–1342, 2000. doi: 10.1137/S0895479898346995. URL https://doi.org/10.1137/S0895479898346995. pages

[38] Evrim Acar, Tamara G. Kolda, and Daniel M. Dunlavy. All-at-once optimization for coupled matrix and tensor factorizations. *CoRR*, abs/1105.3422, 2011. pages

[39] Evrim Acar, Evangelos E. Papalexakis, Gözde Gürdeniz, Morten A. Rasmussen, Anders J. Lawaetz, Mathias Nilsson, and Rasmus Bro. Structure-revealing data fusion. *BMC Bioinformatics*, 15(1):239, Jul 2014. ISSN 1471-2105. doi: 10.1186/1471-2105-15-239. URL https://doi.org/10.1186/1471-2105-15-239. pages

[40] Byungsoo Jeon, Inah Jeon, Lee Sael, and U Kang. Scout: Scalable coupled matrix-tensor factorizationalgorithm and discoveries. pages 811–822, 05 2016. doi: 10.1109/ICDE.2016.7498292. pages

[41] Age K. Smilde and Johan Westerhuis. Multiway multiblock component and covariates regression models. *Journal of Chemometrics - J CHEMOMETR*, 14:301–331, 05 2000. doi: 10.1002/1099-128X(200005/06)14:33.0.CO;2-H. pages

[42] Arindam Banerjee, Sugato Basu, and Srujana Merugu. *Multi-way Clustering on Relation Graphs*, pages 145–156. doi: 10.1137/1.9781611972771.14. URL https://epubs.siam.org/doi/abs/10.1137/1.9781611972771.14. pages

[43] Yu-Ru Lin, J Sun, Paul Castro, Ravi Konuru, Hari Sundaram, and Aisling Kelliher. Metafac: Community discovery via relational hypergraph factorization with propinquity dynamics. pages 527–536, 01 2009. doi: 10.1145/1557019.1557080. pages

[44] Evrim Acar, Canan Aykut-Bingol, Haluk Bingol, Rasmus Bro, and Blent Yener. Multiway analysis of epilepsy tensors. *Bioinformatics*, 23(13):i10–i18, 2007. doi: 10.1093/bioinformatics/btm210. URL http://dx.doi.org/10.1093/bioinformatics/btm210. pages

[45] Fengyu Cong, Qiu-Hua Lin, Li-Dan Kuang, Xiao-Feng Gong, Piia Astikainen, and Tapani Ristaniemi. Tensor decomposition of eeg signals: A brief review. *Journal of Neuroscience Methods*, 248:59 – 69, 2015. ISSN 0165-0270. doi: https://doi.org/10.1016/j.jneumeth.2015.03.018. URL http://www.sciencedirect.com/science/article/pii/S0165027015001016. pages

[46] Otavio G. Lins, Terence W. Picton, Patrick Berg, and Michael Scherg. Ocular artifacts in recording eegs and event-related potentials ii: Source dipoles and source components. *Brain Topography*, 6(1):65–78, Sep 1993. ISSN 1573-6792. doi: 10.1007/BF01234128. URL https://doi.org/10.1007/BF01234128. pages

[47] D. Puthankattil Subha, Paul K. Joseph, Rajendra Acharya U, and Choo Min Lim. Eeg signal analysis: A survey. *J. Med. Syst.*, 34(2):195–212, April 2010. ISSN 0148-5598. doi: 10.1007/s10916-008-9231-z. URL http://dx.doi.org/10.1007/s10916-008-9231-z. pages

[48] Nicole Ille, Patrick Berg, and Michael Scherg. Artifact correction of the ongoing eeg using spatial filters based on artifact and brain signal topographies. *Clin. Neurophysiol.*, 19:113–124, 05 2002. doi: 10.1097/00004691-200203000-00002. pages

[49] Peter Welch. The use of fast fourier transform for the estimation of power spectra: a method based on time averaging over short, modified periodograms. *IEEE Transactions on audio and electroacoustics*, 15(2):70–73, 1967. pages

[50] TACM Claasen and Wolfgang Mecklenbruker. The wigner distributiona tool for time-frequency signal analysispart ii: Discrete time signals. *Philips Jl Research*, 35:276–300, 01 1980. pages

[51] E. Wigner. On the quantum correction for thermodynamic equilibrium. *Phys. Rev.*, 40:749–759, Jun 1932. doi: 10.1103/PhysRev.40.749. URL https://link.aps.org/doi/10.1103/PhysRev.40.749. pages

[52] J Ville. Theorie et applications de la notion de signal analytique. *Cables et transmissions*, 2A:66–74, 1948. pages

[53] H. . Choi and W. J. Williams. Improved time-frequency representation of multicomponent signals using exponential kernels. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37(6):862–871, June 1989. ISSN 0096-3518. doi: 10.1109/ASP.1989.28057. pages

[54] William Williams. Reduced interference distributions: Biological applications and interpretations. *Proceedings of the IEEE*, 84:1264 – 1280, 10 1996. doi: 10.1109/5.535245. pages

[55] Erwin Hennighausen, Martin Heil, and Frank Rsler. A correction method for dc drift artifacts. *Electroencephalography and Clinical Neurophysiology*, 86 (3):199 – 204, 1993. ISSN 0013-4694. doi: https://doi.org/10.1016/0013-4694(93)90008-J. URL http://www.sciencedirect.com/science/article/pii/001346949390008J. pages

[56] Mohammad Ali Tinati and Behzad Mozaffary. A wavelet packets approach to electrocardiograph baseline drift cancellation. *International journal of biomedical imaging*, 2006:97157, 11 2006. doi: 10.1155/IJBI/2006/97157. pages

[57] E. Acar, Y. Levin-Schwartz, V. D. Calhoun, and T. Adali. Tensor-based fusion of eeg and fmri to understand neurological changes in schizophrenia. In *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–4, May 2017. doi: 10.1109/ISCAS.2017.8050303. pages

[58] Alexandre Gramfort, Martin Luessi, Eric Larson, Denis A. Engemann, Daniel Strohmeier, Christian Brodbeck, Lauri Parkkonen, and Matti S. Hmlinen. Mne software for processing meg and eeg data. *NeuroImage*, 86:446 – 460, 2014. ISSN 1053-8119. doi: https://doi.org/10.1016/j.neuroimage.2013.10.027. URL http://www.sciencedirect.com/science/article/pii/S1053811913010501. pages

[59] John Henderson, Steven Luke, Joseph Schmidt, and John Richards. Dataset 3: Natural reading, 2015. URL http://www2.hu-berlin.de/eyetracking-eeg/testdata.html#exampledata3. pages

[60] John Henderson, Steven Luke, Joseph Schmidt, and John Richards. Co-registration of eye movements and event-related potentials in connected-text paragraph reading. *Frontiers in Systems Neuroscience*, 7:28, 2013. ISSN 1662-5137. doi: 10.3389/fnsys.2013.00028. URL https://www.frontiersin.org/article/10.3389/fnsys.2013.00028. pages

[61] Lucas Parra, Clay Spence, Adam Gerson, and Paul Sajda. Recipes for the linear analysis of eeg. *NeuroImage*, 28:326–41, 12 2005. doi: 10.1016/j.neuroimage.2005.05.032. pages

[62] Christopher Torrence and Gilbert P. Compo. A practical guide to wavelet analysis. *Bulletin of the American Meteorological Society*, 79(1):61–78, 1998. doi: 10.1175/1520-0477(1998)079⟨0061:APGTWA⟩2.0.CO;2. URL https://doi.org/10.1175/1520-0477(1998)079<0061:APGTWA>2.0.CO;2. pages

[63] C J. Willmott and K Matsuura. Advantages of the mean absolute error (mae) over the root mean square error (rmse) in assessing average model performance. *Climate Research*, 30:79, 12 2005. doi: 10.3354/cr030079. pages

[64] T. Chai and R. R. Draxler. Root mean square error (rmse) or mean absolute error (mae)? arguments against avoiding rmse in the literature. *Geoscientific Model Development*, 7(3):1247–1250, 2014. doi: 10.5194/gmd-7-1247-2014. URL https://www.geosci-model-dev.net/7/1247/2014/. pages

[65] Scott Makeig, Anthony J. Bell, Tzyy-Ping Jung, and Terrence J. Sejnowski. Independent component analysis of electroencephalographic data. In *Proceedings of the 8th International Conference on Neural Information Processing Systems*, NIPS'95, pages 145–151, Cambridge, MA, USA, 1995. MIT Press. URL http://dl.acm.org/citation.cfm?id=2998828.2998849. pages

[66] A. Hyvrinen and E. Oja. A fast fixed-point algorithm for independent component analysis. *Neural Computation*, 9(7):1483–1492, July 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.7.1483. pages

[67] Rasmus Bro and Henk A.L. Kiers. A new efficient method for determining the number of components in parafac models. *Journal of Chemometrics*, 17(5): 274–286, 5 2003. ISSN 0886-9383. doi: 10.1002/cem.801. pages

[68] C. Battaglino, G. Ballard, and T. Kolda. A practical randomized cp tensor decomposition. *SIAM Journal on Matrix Analysis and Applications*, 39(2):876–901, 2018. doi: 10.1137/17M1112303. URL https://doi.org/10.1137/17M1112303. pages

[69] A. Phan, P. Tichavsk, and A. Cichocki. Candecomp/parafac decomposition of high-order tensors through tensor reshaping. *IEEE Transactions on Signal Processing*, 61(19):4847–4860, Oct 2013. ISSN 1053-587X. doi: 10.1109/TSP.2013.2269046. pages

[70] N. Halko, P. Martinsson, and J. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53(2):217–288, 2011. doi: 10.1137/090771806. URL https://doi.org/10.1137/090771806. pages

[71] Divyansh Manocha. *Tensor Decompositions for EEG*. Imperial College London, 2019. pages

[72] M. Srensen and L. De Lathauwer. Tensor decompositions with block-toeplitz structure and applications in signal processing. In *2011 Conference Record of the Forty Fifth Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*, pages 454–458, Nov 2011. doi: 10.1109/ACSSC.2011.6190040. pages

[73] Andr L.F. de Almeida, Grard Favier, and Joo Cesar M. Mota. Parafac-based unified tensor modeling for wireless communication systems with application to blind multiuser equalization. *Signal Processing*, 87(2):337 – 351, 2007. ISSN 0165-1684. doi: https://doi.org/10.1016/j.sigpro.2005.12.014. URL http://www.sciencedirect.com/science/article/pii/S0165168406001757. Tensor Signal Processing. pages

[74] R. Badeau and R. Boyer. Fast multilinear singular value decomposition for structured tensors. *SIAM Journal on Matrix Analysis and Applications*, 30(3): 1008–1021, 2008. doi: 10.1137/060655936. URL https://doi.org/10.1137/060655936. pages

[75] Martin Weis. *Multi-Dimensional Signal Decomposition Techniques for the Analysis of EEG Data*. PhD thesis, 05 2015. pages

[76] René J. Huster, Stefan Debener, Tom Eichele, and Christoph S. Herrmann. Methods for simultaneous eeg-fmri: An introductory review. *Journal of Neuroscience*, 32(18):6053–6060, 2012. ISSN 0270-6474. doi: 10.1523/JNEUROSCI.0447-12.2012. URL http://www.jneurosci.org/content/32/18/6053. pages

[77] Kianoush Nazarpour, Hamid R. Mohseni, Christian W. Hesse, Jonathon A. Chambers, and Saeid Sanei. A novel semiblind signal extraction approach for the removal of eye-blink artifact from eegs. *EURASIP Journal on Advances in Signal Processing*, 2008(1):857459, Feb 2008. ISSN 1687-6180. doi: 10.1155/2008/857459. URL https://doi.org/10.1155/2008/857459. pages

[78] Fernando Lopes da Silva, Jan Pieter Pijn, and Peter Boeijinga. Interdependence of eeg signals: Linear vs. nonlinear associations and the significance of time delays and phase shifts. *Brain Topography*, 2(1):9–18, Sep 1989. ISSN 1573-6792. doi: 10.1007/BF01128839. URL https://doi.org/10.1007/BF01128839. pages

A — Visual confirmation of the proposed quantification metrics

A small control experiment was conducted to qualitatively confirm the quantification metrics for the performance on datasets with unknown ground truth are as expected. Three cases are considered in this case, one where the optimal number of components (upon the user's discretion) was selected, one where component that did not account for a large proportion of the dataset's variance were selected, and finally one in which too many components were removed. The reconstructions, along with the original time series, for all are visualised in figure A.1.

The optimal extraction, over extraction and under extraction are shown in Figures A.2, A.3 and A.4 respectively. Correlation clearly depicts that channels that are composed of the largest blink components show a much smaller correlation with the reconstructed in the optimal case. This is not seen for the over or under extraction cases. Larger entropy changes were seen for electrodes closer to the eyes, however unlike what was predicted in chapter 5 it does not seem to be a good metric for performance. In frontal electrodes, a negative change in power is almost always observed - as would be expected due to the significant power of blink artifacts (spreading through all components). The largest reduction in power should be seen for electrodes with the largest ocular artifacts, as is seen in the optimal case. The over extraction has removed similar proportions of power in each electrode, whilst an increase in most electrodes is seen in the under extraction case.

The changes in correlation and power depict a clearer quantification of the performance than the entropy measures.

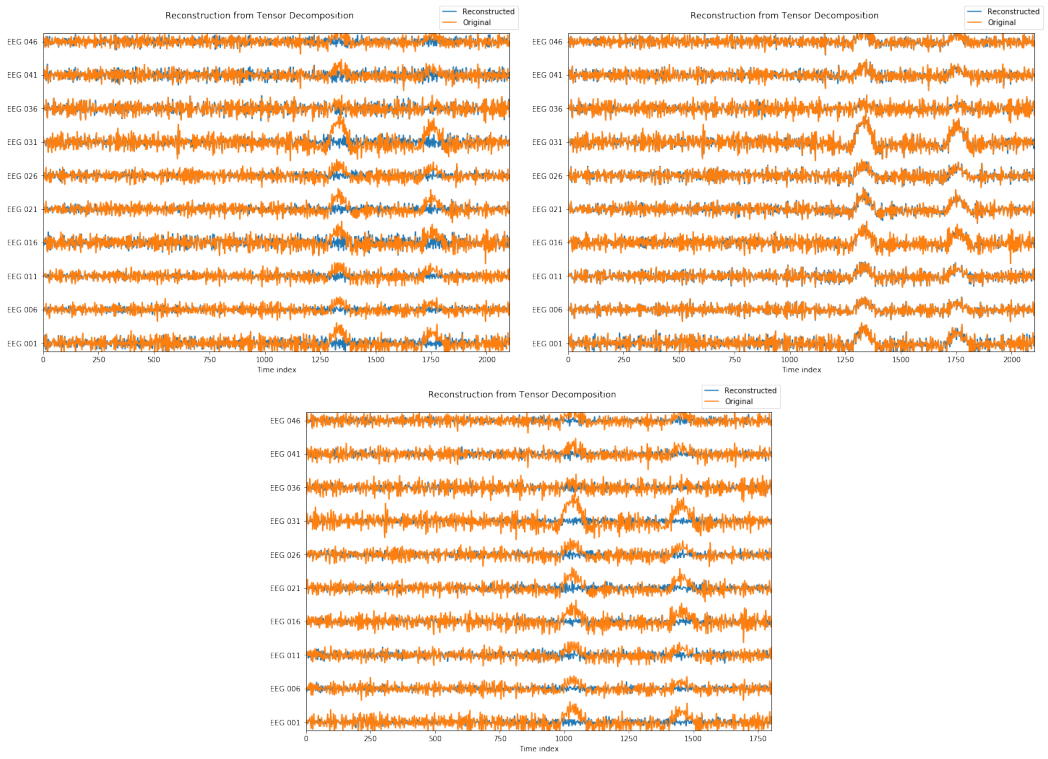

Figure A.1: All artifact extractions were conducted within the same time period, using Tucker decomposition of the same rank.(a) Visually optimally chosen components to remove EEG artifacts. (b) Components that account for smaller variance are selected. (c) The first 60% of the components are chosen.

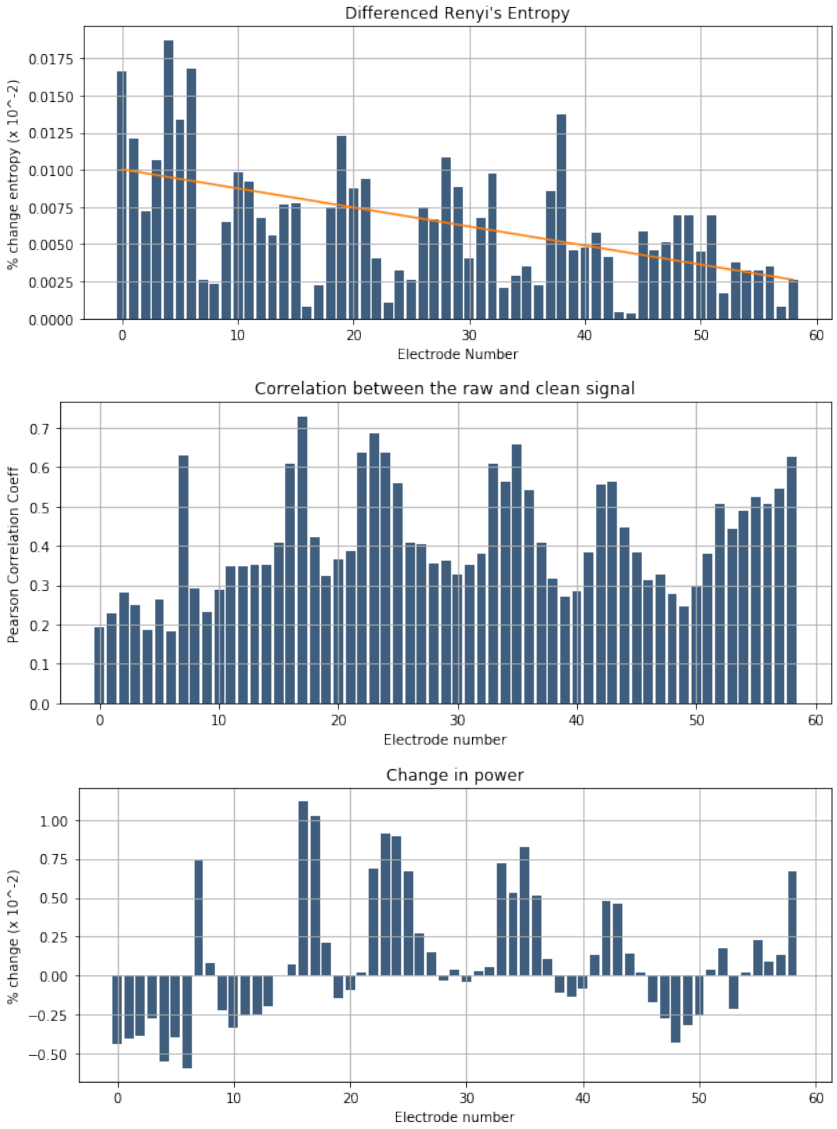

Figure A.2: Optimal extraction of components. (a) Percentage change in Renyi's entropy. (b) Correlation of the origin signal with the reconstructed signal for each electrode. (c) Percentage change in power for each electrode.

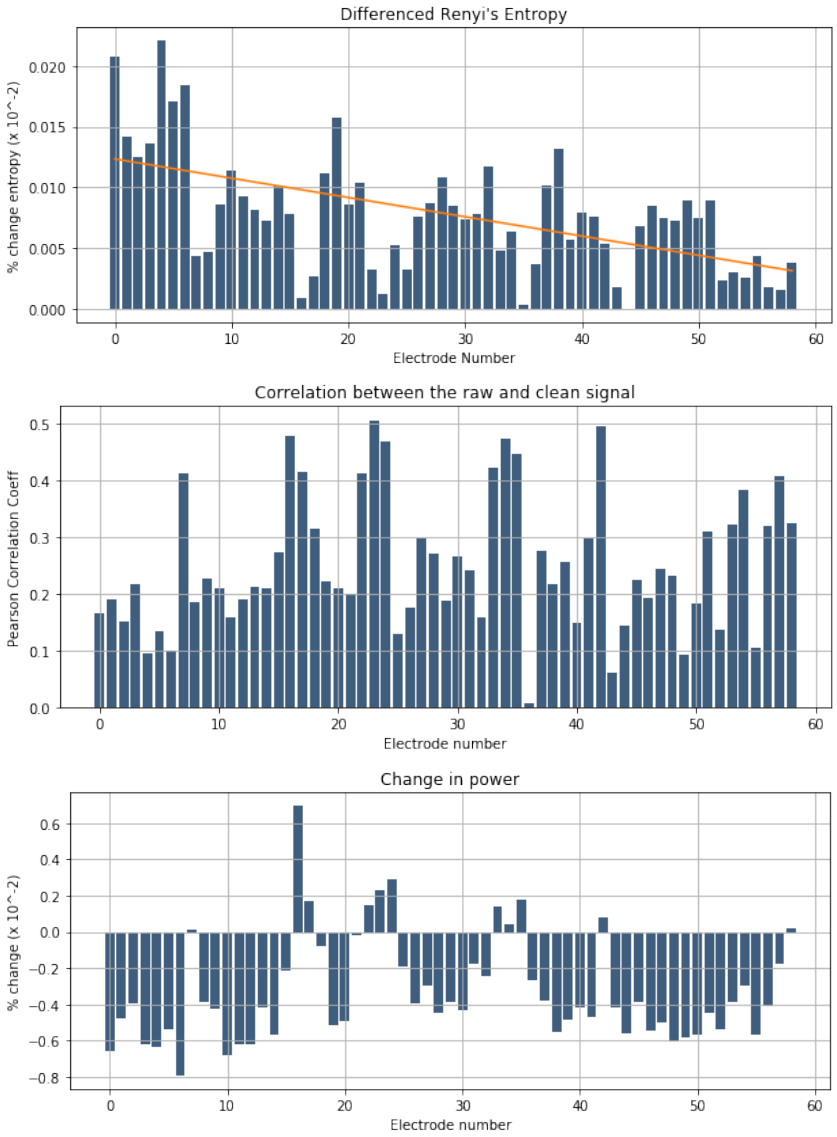

Figure A.3: Over extraction of components. (a) Percentage change in Renyi's entropy. (b) Correlation of the origin signal with the reconstructed signal for each electrode. (c) Percentage change in power for each electrode.

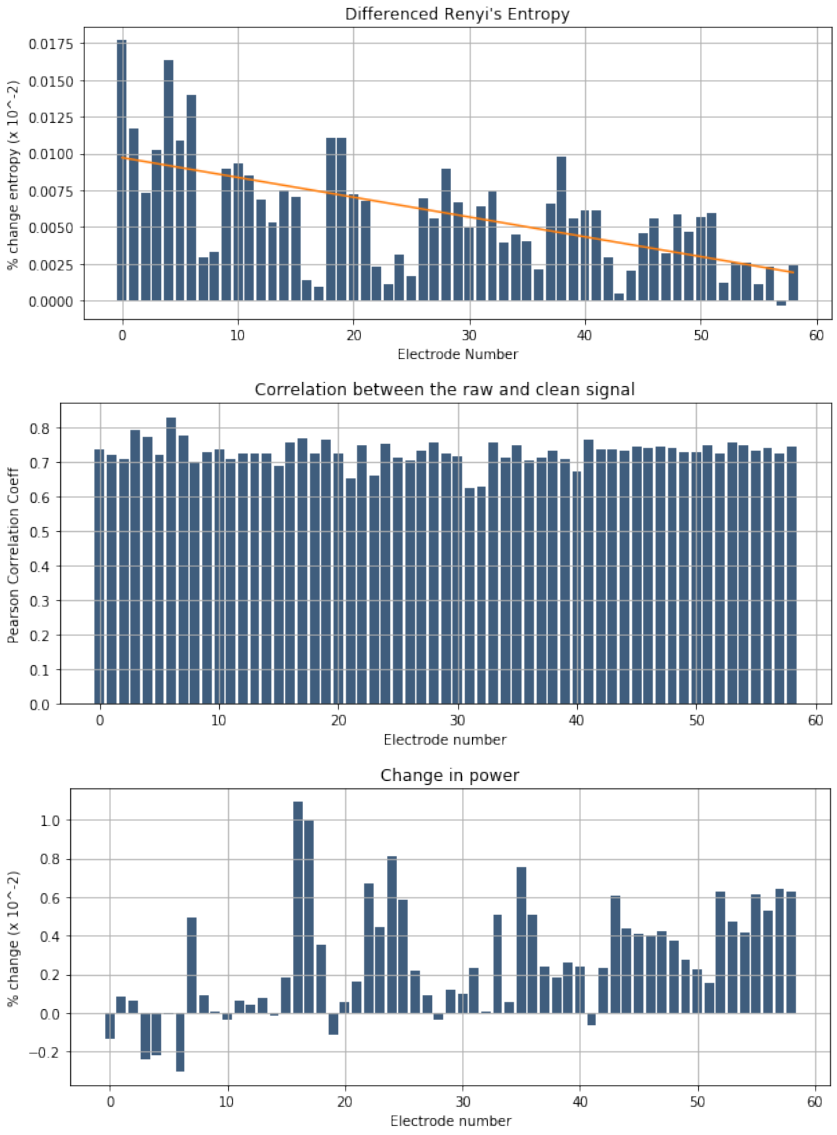

Figure A.4: Under extraction of components. (a) Percentage change in Renyi's entropy. (b) Correlation of the origin signal with the reconstructed signal for each electrode. (c) Percentage change in power for each electrode.

B — Visual analysis of real datasets

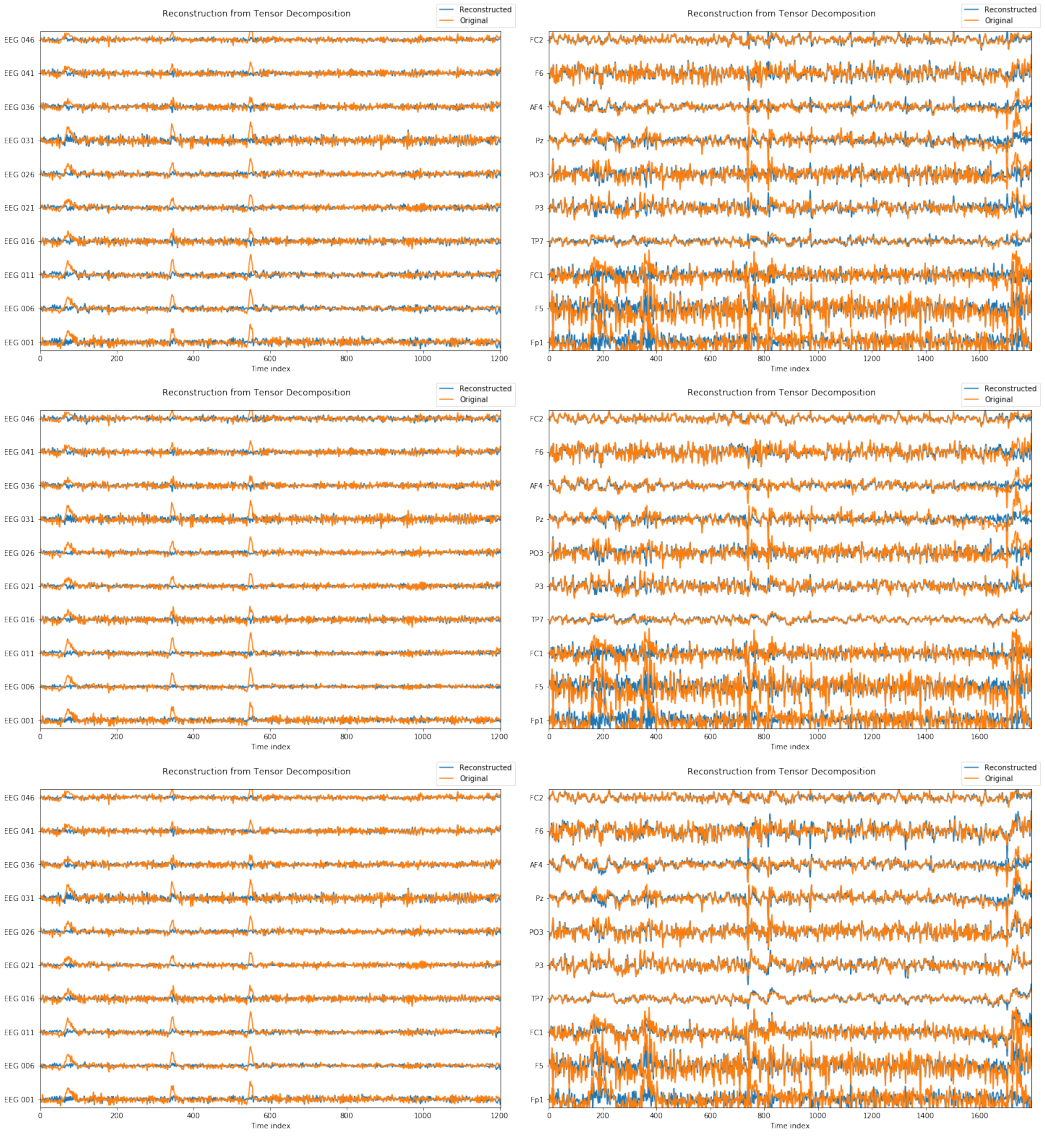

Figure B.1: Reconstructed signals. The results of the MNE dataset are depicted on the right, whilst the right has the reading dataset. (a) Parafac (b) Tucker (c) Parafac2

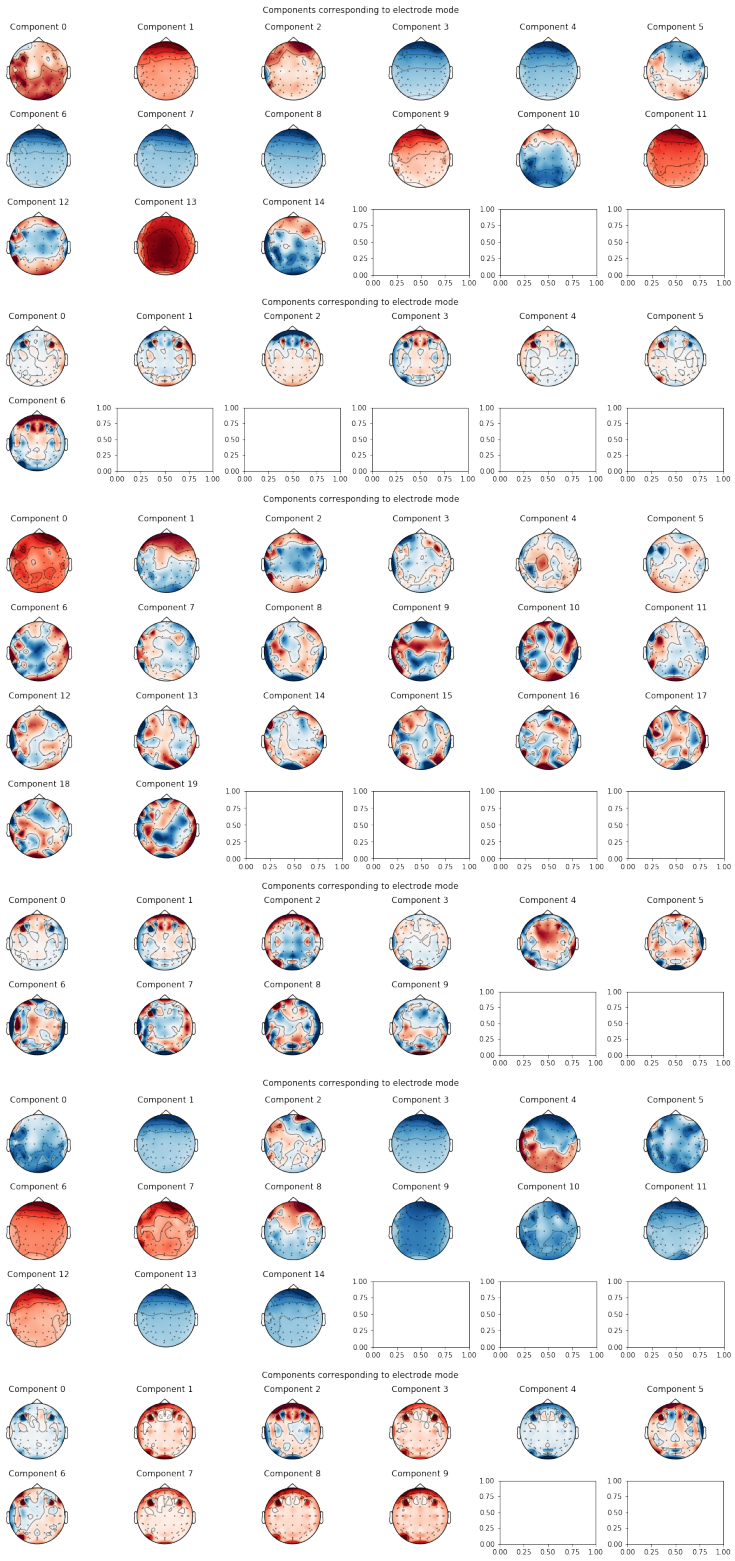

Figure B.2: Extracted components from the decomposition method on the same dataset with optimal number of components. The results of the MNE dataset are depicted on the right, whilst the right has the reading dataset. (a) Parafac on mne (b) Parafac on reading (c) Tucker on mne (d) Tucker on reading (e) Parafac2 on mne (f) Parafac2 on reading