Bayesian Dictionary Learning for EEG Source Identification

Trine Nyholm Kragh & Laura Nyrup Mogensen
Mathematical Engineering, MATTEK

Master's Thesis

Mathematical Engineering
Aalborg University
http://www.aau.dk

AALBORG UNIVERSITY
STUDENT REPORT

Title:
Bayesian Dictionary Learning for EEG Source Identification

Theme:

Project Period:
Fall Semester 2019
Spring Semester 2020

Project Group:
Mattek9b

Participant(s):
Trine Nyholm Kragh
Laura Nyrup Mogensen

Supervisor(s):
Jan Østergaard
Rasmus Waagepetersen

Copies: 1

Page Numbers: ??

Date of Completion:
May 1, 2020

Abstract:

Here is the abstract

Preface

Here is the preface. You should put your signatures at the end of the preface.

Aalborg University, May 1, 2020

Trine Nyholm Kragh
<trijen15@student.aau.dk>

Laura Nyrup Mogensen
<lmogen15@student.aau.dk>

Contents

Introduction

The problem addressed throughout this thesis arise from the increasing use of electroencephalographic measurements for a wide range of scientific purposes, especially within the medical field. An electroencephalography captures electric signals caused by activity within the brain. The signals from the brain is recorded over time by multiple sensors placed on the scalp. One essential issue concerning an electroencephalography is to extract the exact sources of the captured brain activity. This is of interest when studying correlation among activities in different parts of the brain, referred to as functional integration. The recorded signal from one sensor is basically a mixture of electric signals released from a various number of active neurons within the brain, forming one or several sources. Furthermore, this mixture is distorted as it travels through the scalp. The need for source extraction is confirmed by studies showing how analysis performed on electroencephalographic measurements differs significantly from similar analysis performed directly on the original source[13].

Considering this issue of source extraction from a mathematical perspective the electroencephalographic measurements can be modelled by a linear system of equations, from which it is possible to extract a limited number of sources under certain conditions. However, it is a general acknowledged issue that the true number of sources is unknown. The task complexity of extracting the sources from the linear system is increased in cases where the number of sources exceeds the number of sensors providing measurements.

This thesis explores a state of the art mathematical method for source extraction, embracing the case of more sources than sensors. Overall this method, published in 2015, consist of two steps, that is finding receptively the mixture the signals have undergone and then extracting the source signals. The two steps originates from two different approaches considering the mathematical orientation. The main goal of the thesis is to explore and unite the necessary theory into one algorithm. The practical aspect will include an implementation of the algorithm to be tested on new electroencephalographic measurements with the purpose of supporting(?) the current results. Furthermore the problem is connected to an current application within the hearing aid industry. Here the intention is to reduce the amount of energy spent by the hearing aid user. Basically, this is attempted by identifying the listening direction

intended by the user, from analysis of the active sources measured on the user. In this thesis the number of active sources are sought related to the amount of energy used by the hearing aid user. This includes considerations upon the issue of the true number of active sources being unknown.

skal opdateres

(følgende kan skrives bedre hvis ikke det skal være et andet sted)The thesis consist of a motivational part introducing electroencephalography and the potential use within research especially in the hearing aid industry. Furthermore, existing literature considering different mathematical approaches for source extraction are examined. The Motivational part is concluded by the problem statement specifying the objective of the thesis. Next is the theoretical part. The system model is specified and the solution approach are presented. The necessary theory are introduced leading to the state of the art algorithms for source extraction. The theoretical part is followed by implementation and test of the algorithm for verification. Next is.. Finally discussion and conclusion upon the achieved results are presented followed by a consideration upon further studies.

Chapter 1

Motivation

This chapter accounts for the motivation behind source extraction from an Electroencephalography (EEG). The concept of EEG is introduced along with current applications. The potential and importance of source extraction are considered and related to the hearing aid industry. The commonly applied mathematical model for EEG measurements is presented. Currently applied methods for source extraction are considered leading to a presentation of the current state of the art methods which succeeds to overcome the limitations of previous methods. Lastly the objective of this thesis is specified.

1.1 Introduction to EEG Measurements

EEG is an imaging technique used within the medical field. EEG is measuring electric signals on the scalp, caused by brain activity. The human central nerve system consist of various nerve cells connecting the neurons within the brain. Nerve cells respond to certain stimuli, for instance a physical stimuli, and transmit informations between neurons. Generally speaking these activities induce local currents that are transferred throughout the nerve system. Several nearby simultaneous activations result in local potential fields, referred to as one signal *source*[19]. EEG measurements are provided by a number of metal electrodes, referred to as sensors, carefully placed on the human scalp. Each sensor reads the present electrical signals over time. For the source signal to reach a sensor it has to penetrate the skull, skin and several other thin layers of biological tissue. This causes an unknown distortion and reduction of a signal. It is most likely that the measurement of one sensor is a sum of multiple signals from different sources. Nor is the range of a single sensor separated from the other sensors. Thus the same signal can easily be measured by two or more sensors. The process of distorsion and mixing of signals is called volume conduction [19, p. 68] [20]. From this it is clarified that EEG measurements is a mixture of fluctuating electrical signals originating from brain activities. Due to the mixing and the nature of the signals the

true number of sources is generally considered unknown[19]. Furthermore, EEG is a subject for interfering noise. Noise signals can occur in the measurements resulting from physical movement of e.g. eyes and jawbone [22]. The concept of volume conduction is sought illustrated on figure 1.1.

The source signals are classified within four groups according to the dominant frequency. The delta wave (0.5 – 4 Hz) is observed from infants and sleeping adults, the theta wave (4 – 8 Hz) is observed from children and sleeping adults, the alpha wave (8 – 13 Hz) is the most extensively studied brain rhythm, which is induced by an adult laying down with closed eyes. Lastly, the beta wave (13 – 30 Hz) is considered the normal brain wave for adults, associated with active thinking, active attention or solving concrete problems [19, p. 11]. An example of EEG measurements within the four categories is illustrated by figure 1.2.

does this comply to the non-gaussian assumption of ICA?

Generally, the distribution of EEG measurements of multiple sensors are considered multivariant Gaussian[19, p. 50]. Though the mean and covariance properties generally changes over time. Therefore EEG measurements are considered quasistationary i.e. stationary only within small intervals. This motivates the need for segmentation of the EEG measurements to achieve signals with similar characteristics.

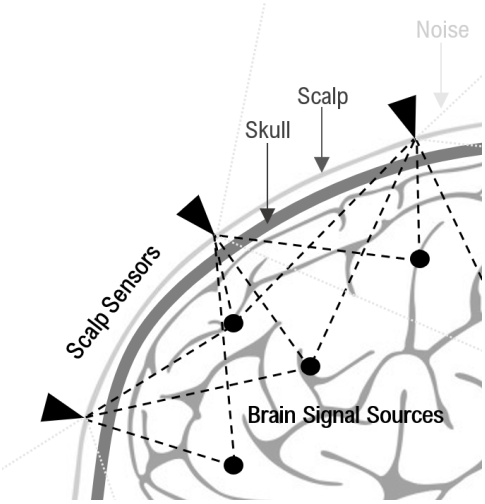

Figure 1.1: Illustration of volume conduction

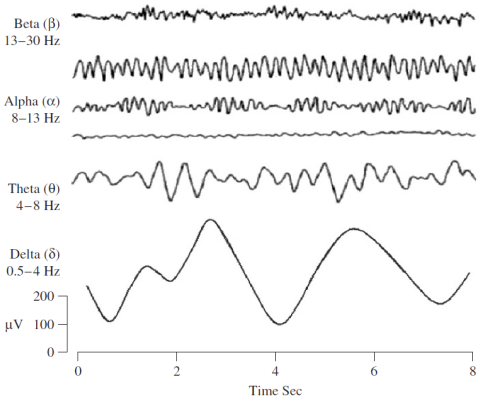

Figure 1.2: Example of time dependent EEG measurements within the four defined categories, source: [19]

1.1.1 Application

EEG performed on humans and animals have a great number of applications with both clinical and research purposes. Examples of clinical applications covers diagnosis and management of neurological disorders such as epilepsy and monitor alertness regarding coma or brain death. EEG capitalizes on the procedure being non-invasive

and fast. Neural activity can be measured within fractions of a second after a stimuli has been provided. These advantages contributes to the wide range of applications within research of the neural processes involved in or resulting from actions, emotions or cognition. Today such neural research are use in many different fields [22, p. 4]. The hearing aid industry is one example where this research is highly prioritized. At Eriksholm research center, which is a part of the hearing aid manufacturer Oticon, cognitive hearing science is a research area within fast development [21]. One main purpose at Eriksholm is to make it possible for a hearing aid to identify the user-intended sound source from real time EEG measurements and thereby exclude noise from elsewhere [2] [6]. It is essentially the well known but unsolved cocktail problem which is sought improved by use of EEG. This is where EEG and occasionally so called in-ear EEG is interesting. In conjunction with the technology of beamforming it is possible for a hearing aid to receive only signals from a specific direction.

Over the past two decades, functional integration has become an area of interest regarding EEG research [12]. Within neurobiology functional integration refers to the study of the correlation among activities in different regions of the brain. In other words, how do different parts of the brain work together to process information and conduct a response [13]. For this purpose separation and localization of the original sources which contribute to the EEG measurement is of interest. An article from 2016 [20] points out the importance of performing analysis regarding functional integration at source level rather than at EEG level. It is argued through experiments that analysis at EEG level does not allow interpretations about the interaction between sources. This emphasize a potential for improving results within a wide range of EEG research, if the original active sources can be extracted from a specific EEG measurements.

1.1.2 Modelling

Consider the issue of extracting the activated sources from EEG measurements on the scalp. A known approach is to model the observed data by a linear system

$$\mathbf{y} = \mathbf{A}\mathbf{x}.$$

The vector $\mathbf{y} \in \mathbb{R}^M$ is the EEG measurement of one time sample containing M sensor measurements. $\mathbf{x} \in \mathbb{R}^N$ is the corresponding N sources within the brain. The non-zero entries of $\mathbf{x}$ represent the active sources at the time of the measurement. $\mathbf{A} \in \mathbb{R}^{M \times N}$ is an unknown projection/transformation(?) matrix, also referred to as the mixing matrix resembling the volume conduction. The i-th column of $\mathbf{A}$ represents the relative projection weights from the i-th source to every sensor [5]. Representing one time sample the linear system is in general referred to as a single measurement vector model. It it only the measurement vector $\mathbf{y}$ that is known hence it is not possible to solve the linear system with respect to $\mathbf{x}$ using basic linear algebra.

The task in this case is to identify both $\mathbf{A}$ and then $\mathbf{x}$, given the measurement vector $\mathbf{y}$. This problem is referred to as the inverse problem of EEG. Finding $\mathbf{x}$ from the inverse problem is referred to as source separation and localization. Separation is to find the signal of each active source and localization is to place each active source signal at the right position within the source vector of dimension N, where N is the maximum number of sources to be active .

der er vel ikke N aktive source her?

Independent Component Analysis (ICA) is one commonly applied method to solve the inverse problem of EEG [16], [15]. ICA is a technique to find the matrix $\mathbf{A}$ such that the column wise elements of $\mathbf{X}$ is statistically independent. Thus statistical independence between the active sources is the essential assumption, which in the case of EEG are considered valid due to the volume conduction being effectively instantaneous [15, p. 3]. Application of ICA has shown great results regarding source separation of high-density EEG. However, a significant flaw to this method is that the EEG measurements are only separated into a number of sources that is equal to or less than the number of sensors [3]. Meaning that the EEG inverse problem can not be solved when it forms an under-determined system, which is the case when the maximum number of unknown sources N exceeds the number of sensors M. Such assumption undermines the reliability and usability of ICA, as the number of active sources easily exceed the number of sensors [5]. This is especially a drawback when low-density EEG are considered. Low-density EEG measurements are collected from equipment with less than 32 sensors, increasing the changes of M being less that N. However, improved capabilities of low-density EEG devices are desirable due to their relative low cost, mobility and ease to use.

This argues the importance of considering the inverse problem of EEG in the under-determined case where $N > M$. In the next section existing work considering the under-determined inverse problem of EEG is investigated further.

1.2 Related Work and Our Objective

As mentioned above ICA is a solid method for source separation in the case where separation into a number of sources equal to the number of sensors is adequate. The issue occurs in cases where the number of sources N exceeds the number of sensors M. To overcome this issue an extension of ICA was suggested, referred to as the ICA mixture model [3]. Instead of identifying one overcomplete mixing matrix $\mathbf{A} \in \mathbb{R}^{M \times N}$ this approach learns N_{model} different mixing matrices $\mathbf{A}_i \in \mathbb{R}^{M \times M}$, to make computations more tractable. This method was further adapted into the Adaptive Mixture ICA (AMICA) which showed successful results regarding identification of more sources than sensors [18]. However, these successful results relies on the assumption that no more than M out of N possible sources is simultaneously active. That is explicit that the source vector of dimension N has at most M non-zero entries. This assumption is still an essential limitation to the frame work,

especially when considering low-density EEG. Other types of ICA algorithms for under-determined systems have been proposed, without overcoming the limitation of jointly active sources exceeding the number of sensors.

In 2015 O. Balkan et. al. suggested a new approach also targeting the identification of more active sources than sensors regarding EEG measurements. One method is proposed for learning $\mathbf{A}$ from $\mathbf{y}$ [3] and a different method is proposed for finding $\mathbf{x}$ given $\mathbf{y}$ and $\mathbf{A}$ [4].

To learn $\mathbf{A}$ the suggested method, referred to as Cov-DL, is a covariance-domain based dictionary learning algorithm. The method is based upon theory of dictionary learning and compressive sensing. Which dictates a framework for solving an under-determined system when $\mathbf{x}$ contains a sufficiently amount of zeros. This is similar to the constraint of ICA. However, to overcome this the point is to transfer the EEG measurements into the covariance domain. In the covariance domain a higher dimensionality can be achieved compared to the original EEG sensor domain with dimension M. The transformation can be done when assuming a linear volume conduction and uncorrelated sources. As a result the theory of compressive sensing is found to apply to the covariance domain, allowing to learn $\mathbf{A}$ by dictionary learning – even in the case where the active sources exceeds the number of measurements.

The Cov-DL algorithm stands out from other straight forward dictionary learning methods as it does not relay on the sparsity of active sources. This is an essential advantage when low-density EEG is considered. Cov-DL was tested and found to outperform AMICA [3]. As mentioned, the Cov-DL algorithm only learns the mixing matrix $\mathbf{A}$, resembling the volume conduction.

is it okay to mention sparsity here for the first time?

For the purpose of recovering $\mathbf{x}$, from $\mathbf{y}$ and $\mathbf{A}$, a multiple measurement sparse Bayesian learning (M-SBL) algorithm is proposed. This method is also targeting the case of more active sources than sensors.The method was proven to outperform the previously used algorithms, even when the defined recovery conditions regarding the found mixing matrix $\mathbf{A}$ was not fulfilled [4].

yderligere beskrivelse nødvedig? eller fjerne lidt fra Cov måske, da det samme lidt kommer i kap 3?

One drawback, which is not fully covered in the referred literature, is that the two methods rely on the number of active sources being known. In practise this is not the case. Hence an estimation of the number of active sources has to be considered for the algorithm to be useful in practice. To address this issue a simple approach is to optimise the result with respect to the number active source, provided that some prior assumption of the expected result can be made.

The two state of the art methods resulting in source separation and localization will make the foundation of this thesis. Our aim is to investigate and fully understand the two methods in order to implement and test a joint algorithm – recovering the original sources $\mathbf{x}$ from the measurements $\mathbf{y}$, when the number of active sources exceeds the number of measurements. Secondary it is of interest to consider the practical application of the algorithm, for instance within a hearing aid as described

in section 1.1. As mentioned, the number of active sources is in general unknown in practise thus it is first of all an estimation of the number of active sources which is of interest for practical use of the algorithm. For this we want to investigate whether it is possible to estimate the number of active sources, through optimization.

Chapter 2

Problem Statement

EEG scalp measurements, a mixture of fluctuating electrical signals originating from brain activities and noise, due to distorting elements such as scalp and biological tissues, can be described as a linear system

$$\mathbf{Y} = \mathbf{AX}.$$

$\mathbf{Y}$ is the EEG scalp measurements measured from M sensors placed on the scalp, $\mathbf{A}$ is the mixing of the electrical signals denoted as the mixing matrix and $\mathbf{X}$ are the N original electrical signals, denoted as sources. Only the EEG measurements $\mathbf{Y}$ is known and it is of interest to identify the mixing matrix $\mathbf{A}$ and hereby the original sources $\mathbf{X}$. The original sources have been shown significant for practical use compared to the raw EEG scalp measurements. Especially, the under-determined case with more sources than sensors is of interest, resulting from low-density EEG devices which is beneficial due to low cost and easy application. In the linear algebraic sense an under-determined linear system have infinitely solution provided a solution exists and is therefore difficult to solve. Two state of the art methods are seen to solve the issue with success, the covariance-domain dictionary learning (Cov-DL) and multiple sparse Bayesian learning (M-SBL) algorithms. The Cov-DL algorithm recovers the mixing matrix from the given measurements $\mathbf{Y}$ while the M-SBL algorithm localised and identify the sources from the recovered mixing matrix and measurements. By combining the two state of art methods into one this could solve the inverse EEG problem – the identification of $\mathbf{A}$ and $\mathbf{X}$ given $\mathbf{Y}$. However, the algorithms used the knowledges of the number of activations within the sources as this is a unknown variable in practice. Hence a modification of the combined state of art methods is sought to increase the potential for practical use.

This motivates the following problem statement.

Based on state of the art method, how can we reproduce the recovering of original sources of brain activity from the EEG inverse problem, in the under-determined

case, and how can this be modified to increase the potential of practical use such as the unknown brain activity?

From the problem statement the following sub-questions is established for clarification.

- Can we reproduce the Cov-DL algorithm to estimate a mixing matrix $\mathbf{A}$ from a over-complete EEG inverse problem with synthetic and realistic EEG scalp measurements?
- Can we reproduce the M-SBL algorithm to estimate a source matrix $\mathbf{X}$ from a over-complete EEG inverse problem with synthetic and realistic EEG scalp measurements?
- How can the number of active sources be estimated, based only on the EEG scalp measurements?

Chapter 3

System Model

Through this chapter a model representing the EEG measurements is specified. Along with the model different terminologies is introduced and described for further use in this thesis. At last the solution approach for estimating the model variables(rather than parameters right?) is described, setting the outline of the remaining chapters of the thesis.

3.1 System of Linear Equations

Let $\mathbf{y} \in \mathbb{R}^M$ be some vector. By basic linear algebra $\mathbf{y}$ can be described as a linear combination of a coefficient matrix $\mathbf{A} \in \mathbb{R}^{M \times N}$ and some scalar vector $\mathbf{x} \in \mathbb{R}^N$ such that

$$\mathbf{y} = \mathbf{A}\mathbf{x}, \tag{3.1}$$

Let $\mathbf{y}$ and $\mathbf{A}$ be known, then 3.1 makes a system of M linear equations with N unknowns, referred to as a linear system.

To solve the linear system 3.1 with respect to $\mathbf{x}$ one must look at the three different cases that can occur, depending on the relation between the number of equations M and the number of unknowns N. For $M = N$, the system has one unique solution, provided that a solution exist. If the square coefficient matrix $\mathbf{A}$ has full rank the solution can be found by inverting $\mathbf{A}$.

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{y}.$$

For $M > N$ the system is over-determined, having more equations than unknown. In general there is no solution to an over-determined system. An/The exception occur when the system contains a sufficient amount of linearly dependent equations. For $M < N$ the system is under-determined, having fewer equations than unknowns. There exist infinitely many solutions to an under-determined system, provided that one solution exist[8, p. ix].

note: skal vi nævne løsnings metoder for under-determined system her?

Consider now $\mathbf{y} \in \mathbb{R}^M$ as the observed measurements provided by M EEG sensors at time t. The linear system 3.1 is then considered as a single measurement vector (SMV) model. Modelling the EEG measurements by the SMV model embody the following interpretations, based on chapter 1. Remember from chapter 1 that EEG measurements basically is a mixture of original brain signals affected by volume conduction. $\mathbf{x}$ is seen as the original brain signal sources, each entry representing the signal of one source. Thus, $\mathbf{x} \in \mathbb{R}^N$ is referred to as the source vector. N is considered the maximum number of sources, however zero-entries may occur. Let k denote the number of non-zero entries in $\mathbf{x}$, referred to as the active sources at time t. The projection matrix $\mathbf{A}$, referred to as the mixing matrix, models the volume conduction by mapping the source vector from $\mathbb{R}^N$ to $\mathbb{R}^M$, where M is the number of sensors hence the dimension of the measurement vector $\mathbf{y}$.

3.2 Multiple Measurement Vector Model of EEG

In practise EEG measurements are sampled over time by a certain sample frequency. Thus multiple EEG measurement vectors are achieved. Let L be the total number of samples. Now the the SMV model is expanded to include L measurement vectors:

$$\mathbf{Y} = \mathbf{AX} + \mathbf{E}, \tag{3.2}$$

now $\mathbf{Y} \in \mathbb{R}^{M \times L}$ is the observed measurement matrix, $\mathbf{X} \in \mathbb{R}^{N \times L}$ is the source matrix, and $\mathbf{A} \in \mathbb{R}^{M \times N}$ is the mixing matrix. Furthermore $\mathbf{E} \in \mathbb{R}^{M \times L}$ is consider an additional noise matrix, to be expected from psychical measurements. The model is now referred to as the multiple measurement vector (MMV) model. As for (3.1) the solution set of the linear system (3.2) depends on the relation between N and M [8, p. 42].

In chapter 1 it is specified that the case of more sources than sensors, $N > M$, is the case of interest in this thesis.

3.2.1 Segmentation

In chapter 1 it is argued that EEG measurements are only stationary within small segments. Hence the following segmentation is considered.

Let f be the sample frequency of the observed EEG measurements $\mathbf{Y}$ and let t be a time interval in seconds determining the duration of one segment. Here s is the segment index. As such the observed EEG measurements can be divided into stationary segments $\mathbf{Y}_s \in \mathbb{R}^{M \times L_s}$, possibly overlapping, where $L_s = tf$ is the number of samples within one segment. For each segment the MMV model (3.2) holds and is rewritten into

$$\mathbf{Y}_s = \mathbf{AX}_s + \mathbf{E}_s, \quad \forall s. \tag{3.3}$$

Due to a segment being stationary it is assumed that each source remains either active or non-active throughout the segment. Thus, $\mathbf{X}_s$, consists of k non-zero rows – the active sources.

In order to characterise the source matrix with respect the amount of non-zero rows the term row sparseness is considered. By common definition the support of the segmented source matrix $\text{supp}(\mathbf{X}_s)$ denotes the index set of non-zero rows of $\mathbf{X}_s$. To count the non-zeros row of a matrix the ℓ_0-norm is defined:

$$\|\mathbf{X}\|_0 := \text{card}(\text{supp}(\mathbf{X})),$$

where the function $\text{card}(\cdot)$ gives the cardinality of the input set. $\mathbf{X}_s$ is said to be k-sparse if it contains at most k non-zeros rows:

$$\|\mathbf{X}_s\|_0 \leq k$$

A model for the EEG measurements is now established. From the model the aim is to recover the source matrix $\mathbf{X}_s \forall s$, which gives us the separated original brain signals as intended by the problem statement. In the next section the solution method is presented and discussed – outlining the remaining chapters of the thesis.

3.3 Solution Method

Denne section er ikke blevet opdateret i forhold til den nye problemformulering. Altså vi mangler at inddrage 'reproducerbarhed'.

It is now justified that the EEG measurements can be modelled by the multiple measurement vector model defined by the system of linear equations (3.3), including an additional noise. By the problem statement cf. chapter 2 the aim is to recover the source vector $\mathbf{X}$, in the case where the number of sensors is less than the number of sources, $M < N$. That is recovering $\mathbf{X}$ from an under-determined linear system. Therefore, the solution must be found in the infinite solution space, provided that one solution exists, thus simple linear algebra can not be used. However, by considering numerical methods such as mathematical optimization it is possible restrict the solution by some constraint and then find the unique optimal solution which respect to a defined cost relative to the solution. The theory of compressive sensing dictates a framework for solving an under-determined system when $\mathbf{X}$ is known to have non-zeros entries. Specifically a unique solution $\mathbf{X}$ can be found when $\mathbf{X}$ is M-sparse, cf. theorem **??** in appendix **??**. When $\mathbf{A}$ is unknown, as it is in the current case, the concept of dictionary learning can be used to determine $\mathbf{A}$, again under the assumption that $\mathbf{X}$ is M-sparse.

As discussed in chapter 1 the aim of this thesis is to overcome the limitation of fewer sources than measurements, which is the limitation of compressive sensing.

A method to overcome this limitation, with respect to learning $\mathbf{A}$, is the Covariance-domain dictionary learning (Cov-DL) method[3], introduced in chapter 1. The

as the relation between **X** and the corresponding δ is unknown(?) det kan vi ikke skrive her kan vi?

method manage to leverage the increased dimensionality of the covariance domain in order to the allow the theory of compressive sensing to apply to an under-determined system. However, this method does only apply to the process of learning **A**, hence a different approach is necessary to recover **X**.

For recovering **X**, given both **Y** and **A**, the method Multiple sparse Bayesian learning (M-SBL), introduced in chapter 1, is considered. A method which ensure that the sparsity holds(?). O. Balkan [4] did also, in 2014, proposed a method which could identify the sources, in the time-domain, by creating a likelihood which ensure the wanted sparsity of the source matrix **X** and controlled by some variance. This method is called multiple sparse Bayesian learning (M-SBL) and takes advantage of a Bayesian approach. In [4] a variance dependent log-likelihood which has been induce by a empirical prior that ensure sparsity of the likelihood has been constructed to be minimised with respect to the variance. From the log-likelihood an estimate for the source matrix **X** is drawn with respect to the support set S which has been influence by variance used in the minimization.

evt. tilføj afslutning der siger hvad der kommer i kapitlerne?

Chapter 4

Covariance-Domain Dictionary Learning

Through this chapter the method Covariance-domain dictionary learning (Cov-DL) is presented in details. Along the presentation of the general method, necessary computational details are derived for the practical solution. The purpose is to recover the mixing matrix $\mathbf{A}$ from the MMV model, derived in chapter 3, in the under-determined case. In the context of compressive sensing the matrix $\mathbf{A}$ in the MMV model is referred to as the dictionary matrix. That is the true mixing matrix is estimated as a dictionary matrix, which can take different forms. This will elaborated further in the section of dictionary learning.

obs! stemmer dette på tværs a kapitlerne, vi siger A bliver estimeret som en dictionary.

Cov-DL is an algorithm proposed by O. Balkan [3], leveraging the increased dimensionality of the covariance domain. The method have shown successful recovering of the mixing matrix $\mathbf{A}$, even in the non-sparse under-determined case with more active sources k than available measurements M, $k \geq M$. In short the algorithm consist of three steps. First the segmented MMV model of the EEG measurements is transformed into the covariance domain. Then, by the increased dimensionality of the covariance domain, it is possible to learn the mixing matrix of the covariance domain, denoted by $\mathbf{D}$, based on the theory of compressive sensing. Here two different cases will appear dependent on the relation between the number of sources N and the found dimension of the covariance domain, which of course depends on the number of measurements M. Lastly, an inverse transformation is performed on the found mixing matrix of the covariance domain $\mathbf{D}$, in order to obtain the wanted mixing matrix $\mathbf{A}$. An important aspect of this method is the prior assumption that the sources within one segment are uncorrelated, that is the rows of $\mathbf{X}_s$ being mutually uncorrelated.

The section is inspired by chapter 3 in [5] and the article [3]. Selected general theory supporting essential parts of the method is elaborated in appendix **??**.

4.1 Covariances Domain Representation

Consider a single sample vector $\mathbf{y}_i$, containing EEG measurements. The covariance of $\mathbf{y}_i$ is to be defined by

$$\mathbf{\Sigma}_{\mathbf{y}_i} = \mathbb{E}[(\mathbf{y}_i - \mathbb{E}[\mathbf{y}_i])(\mathbf{y}_i - \mathbb{E}[\mathbf{y}_i])],$$

where $\mathbb{E}[\cdot]$ is the expected value operator. Assume that all samples vectors $\mathbf{y}_i$ within one segment has zero mean and the same distribution. Then, the observed segmented EEG measurements matrix $\mathbf{Y}_s \in \mathbb{R}^{M \times L_s}$ is to be described in the covariance domain by the sample covariance $\widehat{\mathbf{\Sigma}}$ which is defined as the covariance among the M measurements across the L_s samples. That is a $M \times M$ matrix $\mathbf{\Sigma}_{\mathbf{Y}_s} = [\sigma_{jk}]$ with entries

$$\sigma_{jk} = \frac{1}{L_s} \sum_{i=1}^{L_s} y_{ji} y_{ki}.$$

Using matrix notation the sample covariance of $\mathbf{Y}_s$ can be written as

$$\widehat{\mathbf{\Sigma}}_{\mathbf{Y}_s} = \frac{1}{L_s} \mathbf{Y}_s \mathbf{Y}_s^T.$$

Similar the source matrix $\mathbf{X}_s$ can be described in the covariance domain by the sample covariance matrix.

$$\widehat{\mathbf{\Sigma}}_{\mathbf{X}_s} = \frac{1}{L_s} \mathbf{X}_s \mathbf{X}_s^T = \mathbf{\Lambda}_s + \boldsymbol{\varepsilon}$$

From the assumption of uncorrelated sources within $\mathbf{X}_s$ the sample covariance matrix is expected to be nearly diagonal, thus it can be written as $\mathbf{\Lambda}_s + \boldsymbol{\varepsilon}$ where $\mathbf{\Lambda}_s$ is a diagonal matrix consisting of the diagonal entries of $\widehat{\mathbf{\Sigma}}_{\mathbf{X}_s}$ and $\boldsymbol{\varepsilon}$ is the estimation error[3]. Each segment is then modelled in the covariance domain as

$$\begin{aligned}
\widehat{\mathbf{\Sigma}}_{\mathbf{Y}_s} = \frac{1}{L_s} \mathbf{Y}_s \mathbf{Y}_s^T &= \frac{1}{L_s} \left(\mathbf{A}\mathbf{X}_s + \mathbf{E}_s\right) \left(\mathbf{A}\mathbf{X}_s + \mathbf{E}_s\right)^T \\
\mathbf{Y}_s \mathbf{Y}_s^T &= (\mathbf{A}\mathbf{X}_s)(\mathbf{A}\mathbf{X}_s)^T + \mathbf{E}_s \mathbf{E}_s^T + \mathbf{E}_s (\mathbf{A}\mathbf{X}_s)^T + \mathbf{A}\mathbf{X}_s \mathbf{E}_s^T \\
&= \mathbf{A}\mathbf{X}_s \mathbf{X}_s^T \mathbf{A}^T + \mathbf{E}_s \mathbf{E}_s^T + \mathbf{E}_s \mathbf{X}_s^T \mathbf{A}^T + \mathbf{A}\mathbf{X}_s \mathbf{E}_s^T \\
&= \mathbf{A}(\mathbf{\Lambda}_s + \boldsymbol{\varepsilon})\mathbf{A}^T + \mathbf{E}_s \mathbf{E}_s^T + \mathbf{E}_s \mathbf{X}_s^T \mathbf{A}^T + \mathbf{A}\mathbf{X}_s \mathbf{E}_s^T \\
&= \mathbf{A}\mathbf{\Lambda}_s \mathbf{A}^T + \mathbf{A}\boldsymbol{\varepsilon}\mathbf{A}^T + \mathbf{E}_s \mathbf{E}_s^T + \mathbf{E}_s \mathbf{X}_s^T \mathbf{A}^T + \mathbf{A}\mathbf{X}_s \mathbf{E}_s^T \qquad (4.1) \\
&= \mathbf{A}\mathbf{\Lambda}_s \mathbf{A}^T + \widetilde{\mathbf{E}} \qquad (4.2)
\end{aligned}$$

From (4.1) to (4.2) all terms where noise is included are defined as a united noise term $\widetilde{\mathbf{E}}$. By vector notation (4.2) is rewritten to be vectorized. Because the covariance matrix $\widehat{\mathbf{\Sigma}}_{\mathbf{Y}_s}$ is symmetric it is sufficient to vectorize only the lower triangular parts,

including the diagonal. For this the function $\text{vec}(\cdot)$ is defined to map a symmetric $M \times M$ matrix into a vector of size $\frac{M(M+1)}{2} := \widetilde{M}$ making a row-wise vectorization of its upper triangular part. Furthermore, let $\text{vec}^{-1}(\cdot)$ be the inverse function for devectorisation. This results in the following model

$$\widehat{\mathbf{\Sigma}}_{\mathbf{Y}_s} = \sum_{i=1}^{N} \mathbf{a}_i \mathbf{\Lambda}_{s_{ii}} \mathbf{a}_i^T + \widetilde{\mathbf{E}}$$

$$\begin{aligned} \text{vec}(\widehat{\mathbf{\Sigma}}_{\mathbf{Y}_s}) &= \sum_{i=1}^{N} \text{vec}(\mathbf{a}_i \mathbf{a}_i^T) \mathbf{\Lambda}_{s_{ii}} + \text{vec}(\widetilde{\mathbf{E}}) \\ &= \sum_{i=1}^{N} \mathbf{d}_i \mathbf{\Lambda}_{s_{ii}} + \text{vec}(\widetilde{\mathbf{E}}) \\ &= \mathbf{D}\boldsymbol{\delta}_s + \text{vec}(\widetilde{\mathbf{E}}), \quad \forall s. \end{aligned} \tag{4.3}$$

Here $\boldsymbol{\delta}_s \in \mathbb{R}^N$ contains the diagonal entries of the source sample-covariance matrix $\mathbf{\Lambda}_s$ and the matrix $\mathbf{D} \in \mathbb{R}^{\widetilde{M} \times N}$ consists of the columns $\mathbf{d}_i = \text{vec}(\mathbf{a}_i \mathbf{a}_i^T)$. Note that $\mathbf{D}$ and $\boldsymbol{\delta}_s$ are unknown while $\text{vec}(\widehat{\mathbf{\Sigma}}_{\mathbf{Y}_s})$ is known from the observed data. By this transformation to the covariance domain one segments is now represented by the single measurement model with $\widetilde{M}$ "measurements". It has been shown that this transformed model allows for identification of $k \leq \widetilde{M}$ active sources [17], which is a much weaker sparsity constraint than the original sparsity constraint $k \leq M$. The purpose of the Cov-DL algorithm is to leverage this model to find the dictionary $\mathbf{A}$ from $\mathbf{D}$ and then still allow for $k \leq \widetilde{M}$ active sources to be identified. That is the number of active sources are allowed to exceed the number of observations as intended.

4.2 Determination of the Mixing Matrix

The goal is now to learn first $\mathbf{D}$ and then the associated mixing matrix $\mathbf{A}$. Two methods are considered relying on the relation of M and N. For now the noise vector is ignored.

4.2.1 Under-determined system

When $N > \widetilde{M}$ the transformed model (4.3) makes an under-determined system. This is similar to the original MMV model (3.2) being under-determined when $N > M$. Thus, it is from the theory of compressive sensing again possible to solve the under-determined system if a certain sparsity is withhold. Namely $\boldsymbol{\delta}_s$ being $\widetilde{M}$-sparse. Assuming the sufficient sparsity on $\boldsymbol{\delta}_s$ is withhold it is possible to learn the dictionary matrix of the covariance domain $\mathbf{D}$ by traditional dictionary learning methods applied

to the observations represented in the covariance domain $\text{vec}(\widehat{\mathbf{\Sigma}}_{\mathbf{Y}_s})$ for all segments s.

Dictionary Learning

Within the theory of compressive sensing the matrix $\mathbf{A}$ is referred to as a dictionary matrix, as it determines how a sparse vector $\mathbf{x}$ is transformed to the original non-sparse signal. When the dictionary is not known i prior it is essential how to choose the the dictionary matrix in order to achieve the best recovery, of the sparse vector $\mathbf{x}$ from the measurements $\mathbf{y}$. This is clarified from the proof of theorem **??** in appendix **??**. One choice is a pre-constructed dictionary. In many cases the use of a pre-constructed dictionary results in simple and fast algorithms for reconstruction of $\mathbf{x}$ [10]. However, a pre-constructed dictionary is typically fitted to a specific kind of data. For instance the discrete Fourier transform or the discrete wavelet transform are used especially for sparse representation of images [10]. Hence the results of using such dictionaries depend on how well they fit the data of interest, which is creating a certain limitation.

The alternative option is to consider an adaptive dictionary based on a set of training data that resembles the data of interest. For this purpose learning methods are considered to empirically construct a fixed dictionary which can take part in the application. There exist several dictionary learning algorithms. One is the K-SVD algorithm which was presented in 2006 by Elad et al. and found to outperform pre-constructed dictionaries, when computational cost is of secondary interest [1]. The concept of the K-SVD algorithm is introduced, and the more detailed algorithm is to be found in appendix **??**. Consider the measurement matrix $\mathbf{Y} \in \mathbb{R}^M$ consisting of measurement vectors $\{\mathbf{y}_j\}_j^L$ making a set of L training examples forming a linear system

$$\mathbf{y}_j = \mathbf{A}\mathbf{x}_j.$$

from which one can learn a suitable dictionary $\hat{\mathbf{A}}$, and the sparse representation of the source matrix $\hat{\mathbf{X}} \in \mathbb{R}^N$ with the source vectors $\{\hat{\mathbf{x}}_j\}_j^L$. For a known sparsity constraint k the dictionary learning can be defined by the following optimisation problem.

$$\min_{\mathbf{A},\mathbf{X}} \sum_{j=1}^{L} \|\mathbf{y}_j - \mathbf{A}\mathbf{x}_j\|_2^2 \quad \text{subject to} \quad \|\mathbf{x}_j\|_1 \leq k,\ 1 \leq j \leq L. \tag{4.4}$$

where both $\mathbf{A}$ and $\mathbf{x}_j$ are variables to be determined. Learning the dictionary by the K-SVD algorithm constitute joint solving of the optimization problem with respect to $\mathbf{A}$ and $\mathbf{X}$ respectively. An initial $\mathbf{A}_0 = [\mathbf{a}_0, \cdots, \mathbf{a}_N]$ and the corresponding $\mathbf{X}_0$ is determined. Then, for each iteration an update rule is applied to each column of $\mathbf{A}_0$, that is updating first $\mathbf{a}_j$ and then the corresponding row $\mathbf{x}_i$. More details on

the K-SVD algorithm is found in appendix **??**. The uniqueness of $\mathbf{A}$ depends on the recovery sparsity condition. As clarified earlier in 3.3 the recovery of a unique solution $\mathbf{X}^*$ is only possible if $k < M$ [5].

Application of dictionary learning

By the establishments of an dictionary learning algorithm it is now used to learn the transformed dictionary matrix $\mathbf{D}$ in (4.3). Here the transformed and vectorised measurements $\left\{\text{vec}\left(\hat{\mathbf{\Sigma}}_{\mathbf{Y}}\right), \forall s\right\}$ makes the training dataset. By this note that each segment the original measurement sample constitute only one sample in the covariance domain. Thus the number of training samples depends on the length of a segment. When K-SVD is applied and $\mathbf{D}$ is found it is possible to estimate the mixing matrix $\mathbf{A}$ that generated found $\mathbf{D}$ through the relation

$$\mathbf{d}_j = \text{vec}(\mathbf{a}_j \mathbf{a}_j^T).$$

Here each column is found from the optimisation problem

$$\min_{\mathbf{a}_j} \|\text{vec}^{-1}(\mathbf{d}_j) - \mathbf{a}_j \mathbf{a}_j^T\|_2^2,$$

for which the global minimizer is $\mathbf{a}_j^* = \sqrt{\lambda_j}\mathbf{b}_j$. Here λ_j is the largest eigenvalue of $\text{vec}^{-1}(\mathbf{d}_j)$,

redegørelse for resultatet her skal laves

$$\text{vec}^{-1}(\mathbf{d}_j) = \begin{bmatrix} d_{11} & d_{12} & \cdots & d_{1N} \\ d_{21} & d_{22} & \cdots & d_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ d_{N1} & d_{N2} & \cdots & d_{NN} \end{bmatrix}, \quad j \in [N]$$

and $\mathbf{b}_j$ is the corresponding eigenvector.

By this each column of the mixing matrix $\mathbf{A}$ can be estimated hence it is possible to determine the mixing matrix in the case where the measurements transformed into the covariance domain makes an under-determined system, but the necessary sparsity constraint, $\boldsymbol{\delta}_s$ being $\widetilde{M}$-sparse (instead of M-sparse), is withhold.

4.2.2 Over-determined system

Consider again the measurements represented in the covariance domain (4.3). In the case of $N < \widetilde{M}$ an over-determined system is achieved where $\mathbf{D}$ is high and thin. In general such system is inconsistent. Thus it is not possible to find $\mathbf{D}$ by traditionally dictionary learning methods and different methods most be considered.

tjek, har vi tidligerer nævnt at der teoretisk godt kan være en løsning?

When $N < \widetilde{M}$ it is certain from the model (4.3) that the transformed measurements $\text{vec}(\widehat{\mathbf{\Sigma}}_{\mathbf{Y}_s})$ will live on or near a subspace of dimension N. This subspace is spanned by the columns of $\mathbf{D}$, and is denoted as $\mathcal{R}(\mathbf{D})$. To learn $\mathcal{R}(\mathbf{D})$ without having to impose any sparsity constraint on $\boldsymbol{\delta}_s$ it is possible to use Principal Component

Analysis(PCA). When PCA is applied to the set $\left\{\text{vec}(\widehat{\mathbf{\Sigma}}_{\mathbf{Y}_s}), \forall s\right\}$ a set of N principal components are found. The principal components forms a set of basis vectors $\mathbf{U}$ such that $\mathcal{R}(\mathbf{U}) = \mathcal{R}(\mathbf{D})$. However, this do not imply that $\mathbf{D} = \mathbf{U}$. In the case of two sets of basis vectors spanning the same space, namely $\mathcal{R}(\mathbf{U}) = \mathcal{R}(\mathbf{D})$, the projection operator of the given subset must be unique(need prove here? or is there just one $P : V \to V$ where $V = \mathcal{R}(\mathbf{U}) = \mathcal{R}(\mathbf{D})$).The projection matrix $\mathrm{P} : \mathcal{R}(\mathbf{U}) \to \mathcal{R}(\mathbf{D})$(U?) can be find by considering the projection of an vector $\mathbf{b} \in \mathbb{R}^{\widetilde{M}}$ onto $\mathcal{R}(\mathbf{U})$, that is solving the least squares problem $\|\mathbf{Ax} - \mathbf{b}\|^2$ where $\mathbf{Ax} \subset \mathcal{R}(\mathbf{U})$ and $\mathrm{P} = \mathbf{Ax}$(passer P=Ax her?). When $\mathbf{A}$ has full rank the solution is given by the normal equation

fact: any column in D must be a linear combination of columns in U and visa versa.
Any basis has the same dim(dimension theorem).

$$\begin{aligned}\mathbf{A}^T\mathbf{Ax} &= \mathbf{A}^T\mathbf{b}\\ \mathbf{x} &= (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b}\end{aligned}$$

resulting in

$$\begin{aligned}\mathrm{Pb} = \mathbf{Ax} &= \mathbf{A}(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b}\\ \mathrm{P} &= \mathbf{A}(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\end{aligned}$$

Thus $\mathcal{R}(\mathbf{U})$ and $\mathcal{R}(\mathbf{D})$ having the same projection matrix is true if and only if $\mathbf{D}(\mathbf{D}^T\mathbf{D})^{-1}\mathbf{D}^T = \mathbf{U}(\mathbf{U}^T\mathbf{U})^{-1}\mathbf{U}^T$. Now, remember from the relation between $\mathbf{A}$ and $\mathbf{D}$ that $\mathbf{d}_i = \text{vec}(\mathbf{a}_i\mathbf{a}_i^T)$. From this it is possible to obtain $\mathbf{A}$ through the following optimisation problem

$$\begin{aligned}\min_{\{\mathbf{a}_i\}_{i=1}^N} \|\mathbf{D}(\mathbf{D}^T\mathbf{D})^{-1}\mathbf{D}^T - \mathbf{U}(\mathbf{U}^T\mathbf{U})^{-1}\mathbf{U}^T\|_F^2\\ \text{s.t. } \mathbf{d}_i = \text{vec}(\mathbf{a}_i\mathbf{a}_i^T)\end{aligned} \tag{4.5}$$

where $\mathbf{U}$ results from PCA performed on $\text{vec}(\widehat{\mathbf{\Sigma}}_{\mathbf{Y}_s})$. In the following section the optimization problem is analysed and processed in order to determine a suitable solution method.

4.2.3 Solution to optimization problem

The optimization problem (4.5) consist of an objective function forming a least-square problem with respect to the frobenius norm. That is a convex quadratic objective function. The constraints is a set of quadratic equality constraints. In general it is a thumb rule that non-linear equality constraint are not convex. Due to the constraints not being considered convex the optimization problem does not meet the requirements of a convex optimization problem. Hence the numerical solution methods for convex optimization problems, for which convergence is ensured, does not imply directly.

Due to the nature of the constraints it should be possible to reformulate the objective function to include the constraints into the objective function. That is

construction an unconstrained least-squares problem, which is a special subclass of convex optimization[7].

Let $\mathbf{D} = f(\mathbf{a}_0, \cdots, \mathbf{a}_N)$ then an optimization problem without constraints is achieved and it can be solved be use of for instance the gradient decent method.

description of the exact solution method is missing

4.3 Pseudo Code of the Cov-DL Algorithm

Algorithm 1 Cov-DL

procedure COV-DL($\mathbf{Y}_s$)
  for $s \leftarrow 1, \cdots,$ n_seg **do**
    compute sample covariance matrix $\widehat{\mathbf{\Sigma}}_{\mathbf{Y}_s}$
    $\mathbf{y}_{\text{cov}_s} = \text{vec}(\widehat{\mathbf{\Sigma}}_{\mathbf{Y}_s})$
  end for
  $\mathbf{Y}_{\text{cov}} = \{\mathbf{y}_{\text{cov}_s}\}_{s=1}^{\text{n\_seg}}$
  if $N \geq \widetilde{M}$ **then**
    procedure K-SVD($\mathbf{Y}_{\text{cov}}$)
      returns $\mathbf{D} \in \mathbb{R}^{\widetilde{M}} \times N$
    end procedure
    for $j \leftarrow 1, \cdots, N$ **do**
      $\mathbf{T} = \text{vec}^{-1}(d_j)$
      $\lambda_j \leftarrow \max\{\text{eigenvalue}(\mathbf{T})\}$
      $\mathbf{b}_j \leftarrow$ eigenvector(λ_j)
      $\mathbf{a}_j \leftarrow \sqrt{\lambda_j}\mathbf{b}_j$
    end for
    $\mathbf{A} = \{\mathbf{a}_j\}_{j=1}^{N}$
  end if

  if $N < \widetilde{M}$ **then**
    procedure PCA(vec($\mathbf{\Sigma}_{\mathbf{Y}_s}$))
      returns $\mathbf{U} \in \mathbb{R}^{\widetilde{M} \times N}$
    end procedure
    procedure MIN. $\mathbf{A}$ IN ($\|\mathbf{D}(\mathbf{D}^T\mathbf{D})^{-1}\mathbf{D}^T - \mathbf{U}(\mathbf{U}^T\mathbf{U})^{-1}\mathbf{U}^T\|_F^2$)
      returns $\mathbf{A} = \{\mathbf{a}_j\}_{j=1}^{N}$
    end procedure
  end if
end procedure

4.4 Considerations and Remarks

Through this chapter different theory aspects haven been investigated to create a foundation to present one method to be use in the localisation of the sources from EEG measurements – the recovering of the mixing matrix $\mathbf{A}$ – yet one method is still to be presented. Before the method to recover the source matrix $\mathbf{X}$ from the found mixing matrix $\mathbf{A}$ and EEG measurements $\mathbf{Y}$ will be introduced some considerations and remarks regarding the Cov-DL algorithm must be taken – this will be used in the implementation of the algorithm which will be described in chapter 6.

The length of each segment determined whenever the covariance of the source matrix $\mathbf{X}$ can be described as a diagonal matrix $\mathbf{\Lambda}$. That is a segment of L_s samples becomes stationary and therefore the sources within that segment becomes uncorrelated – the covariance of the source can be described by a diagonal matrix. The number of samples L_s used in one segment affect whenever the segment is stationary or not. This must be taken into account in the preprocessing part of the baseline algorithm when the EEG measurements are divided into segments.

For the Cov-DL algorithm when $\mathbf{D}$ is under-determined a dictionary learning algorithm K-SVD is used to learn the matrix $\mathbf{D}$ and by that an estimate for the mixing matrix $\hat{\mathbf{A}}$. Because of the segmentation the number of samples used in the dictionary learning are reduced remarkably and will affect the learning process. This is another point which must be taken into account in the preprocessing part of the code. To improved the dictionary learning the overlapping of the segments can be look into as each segment will have some similarity and therefore learn towards one direction.

For the Cov-DL algorithm when $\mathbf{D}$ is over-determined the solution tends to be unique when $M < N < \widetilde{M}$ from testing the solution. That is the cost function tends toward a local minima and therefore an unique solution occur in first run of one trial. For the baseline algorithm it would therefore be necessary to include several random initial points when finding the mixing matrix $\mathbf{A}$ for $\mathbf{D}$ being over-determined.

For a general perspective the sources within the source matrix $\mathbf{X}$ must not be constant over time when using the MMV model (3.2) ...

Find lige kilde på dette argument

Chapter 5

Multiple Sparse Bayesian Learning

In this chapter the multiple sparse Bayesian learning (M-SBL) method is described in details.As the method leverage a Bayesian framework the general concept of Bayesian inference is shortly introduced prior to the the M-SBL method, with respect to the model of interest (5.1). The chapter is inspired by [23] and the articles [24], [4].

citat: p. 9(pdf) sparse Bayesian learning (SBL) is an empirical Bayesian approaches, which use a parameterized prior to encourage sparsity through a process called evidence maximization

Consider again the multiple measurement vector (MMV) model for a non-segmented case of EEG measurements

$$\mathbf{Y} = \mathbf{A}\mathbf{X} + \mathbf{E}, \tag{5.1}$$

with measurement matrix $\mathbf{Y} \in \mathbb{R}^{M \times L}$, source matrix $\mathbf{X} \in \mathbb{R}^{N \times L}$ mixing matrix $\mathbf{A} \in \mathbb{R}^{M \times N}$ and noise matrix $\mathbf{E} \in \mathbb{R}^{M \times L}$. Note that $\mathbf{A}$ is known throughout the chapter, as it is found by Cov-DL in chapter 4.

The aim is to recover the source matrix $\mathbf{X}$ in the case of fewer measurements than active sources, $k > M$. In [4] it is proven that exact localization of the active sources can be achieved with M-SBL for $k > M$, when two sufficient conditions are satisfied. The basic approach of M-SBL is to find the support set S providing the non-zeros rows of the source matrix $\mathbf{X}$ which corresponds to localization of the active sources. Finally the value of the localized active sources are estimated.

5.1 Bayesian Inference

General Bayesian statistics builds upon the task of inferring what the model parameters must be, given the model and data. This is centred around Bayes' theorem, which is a posterior distribution of some unobserved variable given some observed variable.

Consider now the current non-segmented MMV model (5.1) within the Bayesian framework, the model parameter – the source matrix $\mathbf{X}$ – is wished estimated given

the measurement matrix $\mathbf{Y}$. Bayes' theorem becomes

$$p(\mathbf{X}|\mathbf{Y}) = \frac{p(\mathbf{Y}|\mathbf{X})p(\mathbf{X})}{p(\mathbf{Y})},$$

where $p(\mathbf{Y}|\mathbf{X})$ is the probability density function of $\mathbf{Y}$ given $\mathbf{X}$, also referred to as a likelihood function, $p(\mathbf{X})$ is a prior distribution of $\mathbf{X}$ and $p(\mathbf{Y})$ is the distribution of $\mathbf{Y}$ serving as a normalizing parameter. By maximizing the posterior distribution $p(\mathbf{X}|\mathbf{Y})$ with respect to $\mathbf{X}$, the maximum a posteriori (MAP) estimate, an estimate for the source matrix can be found as

$$\hat{\mathbf{X}}_{\text{MAP}} = \arg\max_{\mathbf{X}} \frac{p(\mathbf{Y}|\mathbf{X})p(\mathbf{X})}{p(\mathbf{Y})}.$$

hvordan er dette ikke korrekt? 'rasmus'

That is the estimate of $\mathbf{X}$, from the MMV model (5.1), with the highest probability of causing the given variable $\mathbf{Y}$. In the desired case where $M < N$ the MMV model (5.1) makes an under-determined system and potentially an infinitely number of solutions exist with equal likelihoods.

Let the source matrix $\mathbf{X}$ be seen as a variable which is drawn from some distribution $p(\mathbf{X})$, as such it is possible to narrow down the infinitely solution space. Assuming a prior belief that $\mathbf{Y}$ is generated from a sparse source matrix, gives a so-called sparsity inducing prior. That is $\mathbf{X}$ is drawn from some distribution which has a sharp, possibly infinite, spike at zero surrounded by fat tails. A specific example could be a prior distribution $p(\mathbf{X}) \propto \exp(-\|\mathbf{X}\|_0)$ [23, p. 14]. However, for simplicity a Gaussian prior is to prefer. The use of a Gaussian distribution can (almost) be justified if a mixture of two Gaussian distributions are considered such that the variable is drawn from one of the two with equal likelihood. One where the variance of the distribution is close to zero, resembling the narrow spike around the mean at zero. And, one with high variance resembling the fat tails.

giver det mening med den mixture?

Different MAP estimation approaches exists separated by the choice of sparsity inducing prior and optimization method. However, regardless of the approach some problems have shown to occur when using a fixed and algorithm-dependent prior. One issue is the posterior not being sparse enough if a prior is not as sparse, leading to non-recovery. Another issue is that a combinatorial number of suboptimal local solutions can occur. By use of automatic relevance determination (ARD) the problems related to the fixed sparse prior can be avoided [23, p. 20]. The main asset of this alternative approach is the use of an empirical prior. That is an flexible prior distribution which depends on an unknown set of hyperparameters, which is to be learned from the data.

5.1.1 Empirical Bayesian Estimation

Assume the likelihood function $p(\mathbf{Y}|\mathbf{X})$ is Gaussian, with known noise variance σ^2. Due to $\mathbf{Y}$ containing samples from multiple sensors over time each entry of $\mathbf{Y}$ are

independent and equally distributed with likelihood

$$\begin{aligned} p(y_{ij}|x_{ij}) &\sim \mathcal{N}(\mathbf{A}_{i\cdot}\mathbf{x}_{\cdot j}, \sigma^2) \\ &= \frac{1}{\sigma^2\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{y_{ij} - \mathbf{A}_{i\cdot}\mathbf{x}_{\cdot j}}{\sigma}\right)^2\right) \end{aligned}$$

Tidligere definition a likelihood

$$\begin{aligned} p(\mathbf{y}_{\cdot j}|\mathbf{x}_{\cdot j}) &= \mathcal{N}(\mathbf{A}\mathbf{x}_{\cdot j}, \sigma^2\mathbf{I}) \\ &= (2\pi)^{-\frac{M}{2}} |\sigma^2\mathbf{I}|^{-\frac{1}{2}} \exp\left(-\frac{1}{2\sigma^2\mathbf{I}}\|\mathbf{y}_{\cdot j} - \mathbf{A}\mathbf{x}_{\cdot j}\|_2^2\right) \end{aligned}$$

Now the empirical prior is defined by application of ARD. Similar to the entries of $\mathbf{Y}$ each parameter x_{ij} are independent and equally distributed by a Gaussian distribution with zero mean and a variance controlled by an unknown hyperparameter γ_i:

$$p(x_{ij};\gamma_i) \sim \mathcal{N}(0, \gamma_i).$$

Note that every entry of row i is controlled by the same hyperparameter γ_i, that is one source signal over time is controlled by one hyperparameter. By combining the prior of each parameter, the prior of $\mathbf{X}$ is fully specified as follows

$$p(\mathbf{X};\boldsymbol{\gamma}) = \prod_{i=1}^{N} p(\mathbf{x}_{i\cdot};\gamma_i),$$

with the hyperparameter vector $\boldsymbol{\gamma} = [\gamma_1, \ldots, \gamma_N]^T$. Note that one column of the unknown sources $\mathbf{x}_{\cdot j}$ depends on the $\boldsymbol{\gamma}$. The prior can be composed as

$$p\left(\mathbf{x}_{\cdot j};\boldsymbol{\gamma}\right) = \prod_{i=1}^{N} p\left(x_{ij};\gamma_i\right).$$

giver det mening? hvis 5.2 skulle skrive om så skal det hele være produkter ikke?

Combining the prior and the likelihood $p(\mathbf{y}_{\cdot j}|\mathbf{x}_{\cdot j})$ the posterior of the j-th column of the source matrix $\mathbf{X}$ becomes

$$\begin{aligned} p(\mathbf{x}_{\cdot j}|\mathbf{y}_{\cdot j};\boldsymbol{\gamma}) &= \frac{p(\mathbf{y}_{\cdot j}|\mathbf{x}_{\cdot j};\boldsymbol{\gamma})p(\mathbf{x}_{\cdot j};\boldsymbol{\gamma})}{p(\mathbf{y}_{\cdot j}|\boldsymbol{\gamma})} \\ &= \frac{p(\mathbf{y}_{\cdot j}|\mathbf{x}_{\cdot j};\boldsymbol{\gamma})p(\mathbf{x}_{\cdot j};\boldsymbol{\gamma})}{\int p(\mathbf{y}_{\cdot j}|\mathbf{x}_{\cdot j})p(\mathbf{x}_{\cdot j};\boldsymbol{\gamma})\ d\mathbf{x}_{\cdot j}} \\ &\propto p(\mathbf{y}_{\cdot j}|\mathbf{x}_{\cdot j};\boldsymbol{\gamma})p(\mathbf{x}_{\cdot j};\boldsymbol{\gamma}) \qquad (5.2) \\ &\sim \mathcal{N}(\boldsymbol{\mu}_{\cdot j}, \boldsymbol{\Sigma}), \qquad (5.3) \end{aligned}$$

where the denominator is the marginal likelihood of $\mathbf{y}_{\cdot j}$ also referred to as the evidence. The marginalization is elaborated in the next section. The mean and covariance of (5.3) is given as

$$\boldsymbol{\Sigma} = \mathrm{Cov}(\mathbf{x}_{\cdot j}|\mathbf{y}_{\cdot j};\boldsymbol{\gamma}) = \boldsymbol{\Gamma} - \boldsymbol{\Gamma}\mathbf{A}^T\boldsymbol{\Sigma}_y^{-1}\mathbf{A}\boldsymbol{\Gamma}, \quad \forall j = 1, \ldots, L \qquad (5.4)$$

$$\mathcal{M} = [\boldsymbol{\mu}_{\cdot 1}, \ldots, \boldsymbol{\mu}_{\cdot L}] = \mathbb{E}[\mathbf{X}|\mathbf{Y};\boldsymbol{\gamma}] = \boldsymbol{\Gamma}\mathbf{A}^T\boldsymbol{\Sigma}_y^{-1}\mathbf{Y}, \qquad (5.5)$$

where $\mathbf{\Gamma} = \text{diag}(\boldsymbol{\gamma})$ and $\mathbf{\Sigma}_y = \sigma^2\mathbf{I} + \mathbf{A}\mathbf{\Gamma}\mathbf{A}^T$. The derivation of the posterior mean and covariance if found in appendix 5.2.

Let the posterior mean $\mathcal{M}$ serve as the point estimate for the source matrix $\mathbf{X}$. It is clear that row sparsity is achieved whenever $\gamma_i = 0$. From this the posterior must satisfy the following

$$\mathbb{P}(\mathbf{x}_{i\cdot} = \mathbf{0}|\mathbf{Y}; \gamma_i = 0) = 1.$$

This ensures that the posterior mean $\mathcal{M}$ of the i-th row, $\boldsymbol{\mu}_{i\cdot}$, become zero, whenever $\gamma_i = 0$ as desired.

From this it is evident that for estimating the support set of $\mathbf{X}$ it is sufficient to estimate the hyperparameter $\boldsymbol{\gamma}$, from which the support set S can be extracted. Furthermore, the point estimate of $\mathbf{X}$, providing the source signal estimate, is given by $\mathcal{M}$ [23, p. 147]. This leads to the actual M-SBL algorithm for which the aim is to estimate $\boldsymbol{\gamma}$ and the corresponding $\mathcal{M}$.

5.2 Derivation of posterior mean and covariance(put in appendix)

The purpose is here to derive the mean and covariance of the posterior distribution

$$p\left(\mathbf{x}_{\cdot j}|\mathbf{y}_{\cdot j}; \boldsymbol{\gamma}\right) \sim \mathcal{N}(\boldsymbol{\mu}_{\cdot j}, \mathbf{\Sigma}).$$

from (5.3) in section 5.1.1. Let now $\mathbf{x}_{\cdot j} = \mathbf{x}$ and $\mathbf{y}_{\cdot j} = \mathbf{y}$

We have

$$p(\mathbf{x}; \boldsymbol{\gamma}) \sim \mathcal{N}(0, \boldsymbol{\gamma}\mathbf{I})$$

and

$$p(\mathbf{y}|\mathbf{x}) \sim \mathcal{N}(\mathbf{A}\mathbf{x}, \boldsymbol{\sigma}^2\mathbf{I}).$$

Now define one Gaussian random variable

$$\mathbf{z} = \begin{bmatrix} \mathbf{x} \\ \mathbf{y}|\mathbf{x} \end{bmatrix} \in \mathbb{R}^{N+M},$$

then the mean and covariance of $\mathbf{z}$ can be partitioned into

$$\mathbf{\Sigma}_\mathbf{z} = \begin{bmatrix} \mathbf{\Sigma}_{\mathbf{xx}} & \mathbf{\Sigma}_{\mathbf{xy}} \\ \mathbf{\Sigma}_{\mathbf{yx}} & \mathbf{\Sigma}_{\mathbf{yy}} \end{bmatrix} \in \mathbb{R}^{N+M \times N+M}$$

$$\boldsymbol{\mu}_\mathbf{z} = \begin{bmatrix} \boldsymbol{\mu}_\mathbf{x} \\ \boldsymbol{\mu}_{\mathbf{y}|\mathbf{x}} \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{A}\mathbf{x} \end{bmatrix}.$$

Then, from [9], the posterior distribution of $\mathbf{x}$ given $\mathbf{y}$ is defined as

$$\mathbf{\Sigma} = \text{cov}(\mathbf{x}, \mathbf{x}|\mathbf{y}) = \mathbf{\Sigma}_{\mathbf{xx}} - \mathbf{\Sigma}_{\mathbf{xy}}\mathbf{\Sigma}_{\mathbf{yy}}^{-1}\mathbf{\Sigma}_{\mathbf{yx}}$$
$$\boldsymbol{\mu} = \boldsymbol{\mu}_\mathbf{x} + \mathbf{\Sigma}_{\mathbf{xy}}\mathbf{\Sigma}_{\mathbf{yy}}^{-1}(\mathbf{y} - \boldsymbol{\mu}_\mathbf{y})$$

Consider first the conditional covariance. Each covariance within the expression is now found:
The covariance of $\mathbf{x}$ comes directly from the distribution

$$\boldsymbol{\Sigma}_{\mathbf{xx}} = \boldsymbol{\gamma}\mathbf{I}$$

The covariance between $\mathbf{x}$ and $\mathbf{y}|\mathbf{x}$ is found by .....

how to treat $\mathbf{y}$ being conditional here? can't get the right result

$$\begin{aligned}\boldsymbol{\Sigma}_{\mathbf{yx}} &= \mathbb{E}[XY] - \mathbb{E}[X]\mathbb{E}[Y] \\ &= \mathbb{E}[XY] - 0\mathbb{E}[Y] \\ &= \mathbb{E}[XY] \\ &\dots \\ &= \boldsymbol{\gamma}\mathbf{I}\mathbf{A}^T?\end{aligned}$$

Lastly the covariance of $\mathbf{y}|\mathbf{x}$ is similarly found by the definition of conditional covariance as follows

$$\begin{aligned}\boldsymbol{\Sigma}_{\mathbf{yy}} = \operatorname{cov}(\mathbf{y},\mathbf{y}|\mathbf{x}) = \boldsymbol{\Sigma}_{\mathbf{yy}|\mathbf{x}} &= \boldsymbol{\Sigma}_{\mathbf{yy}} - \boldsymbol{\Sigma}_{\mathbf{yx}}\boldsymbol{\Sigma}_{\mathbf{xx}}^{-1}\boldsymbol{\Sigma}_{\mathbf{xy}} \\ &= \boldsymbol{\sigma}^2\mathbf{I} - \boldsymbol{\gamma}\mathbf{I}\mathbf{A}^T(\boldsymbol{\gamma}\mathbf{I})^{-1}\mathbf{A}\boldsymbol{\gamma}\mathbf{I} \\ &= \boldsymbol{\sigma}^2\mathbf{I} - \mathbf{A}\boldsymbol{\gamma}\mathbf{I}\mathbf{A}^T\end{aligned}$$

The resulting conditional covariance becomes

$$\boldsymbol{\Sigma} = \boldsymbol{\gamma}\mathbf{I} - \boldsymbol{\gamma}\mathbf{I}\mathbf{A}^T(\boldsymbol{\Sigma}_{\mathbf{yy}|\mathbf{x}})^{-1}\mathbf{A}\boldsymbol{\gamma}\mathbf{I}.$$

Consider now the mean of $\mathbf{x}$ conditional on $\mathbf{y}$

$$\begin{aligned}\boldsymbol{\mu} &= \boldsymbol{\mu}_{\mathbf{x}} + \boldsymbol{\Sigma}_{\mathbf{xy}}\boldsymbol{\Sigma}_{\mathbf{yy}}^{-1}\left(\mathbf{y} - \boldsymbol{\mu}_{\mathbf{y}}\right) \\ &= 0 + \boldsymbol{\gamma}\mathbf{I}\mathbf{A}^T\left(\boldsymbol{\Sigma}_{\mathbf{yy}}\right)^{-1}\left(\mathbf{y} - \mathbf{A}\mathbf{x}\right)\end{aligned}$$

in the result we have used, the mean of y is subtracted.?

5.3 M-SBL for estimation of X

The M-SBL algorithm is now specified in order to estimate the hyperparameter $\boldsymbol{\gamma}$ and then the corresponding unknown sources $\mathbf{X}$. Due to the empirical Bayesian strategy the unknown variables, making the source matrix $\mathbf{X}$ are integrated out, also referred to as marginalized. By integrating the posterior with respect to the unknown sources $\mathbf{X}$ the marginal likelihood of the observed mixed data $\mathbf{Y}$ is achieved [23, p. 146]

$$\begin{aligned}\mathcal{L}(\boldsymbol{\gamma};\mathbf{Y}) &= \int p(\mathbf{Y}|\mathbf{X})p(\mathbf{X};\boldsymbol{\gamma})\ d\mathbf{X} \\ &= p(\mathbf{Y}|\boldsymbol{\gamma})\end{aligned}$$

when to use ; inst

The resulting marginal likelihood of $\boldsymbol{\gamma}$ is to be maximised with respect to $\boldsymbol{\gamma}$, that is the maximum likelihood estimate (MLE) which is considered the resulting ARD based M-SBL cost function. The $-2\log(\cdot)$ transformation is applied in order for the cost function to be minimized, and factors not depending on $\mathbf{Y}$ is removed. Resulting in the following log likelihood.

ops på L faktor i sidste linje..

$$\begin{aligned}\ell(\boldsymbol{\gamma};\mathbf{Y}) &= -2\log(p(\mathbf{Y};\boldsymbol{\gamma})) \\ &= -2\log\left(2\pi^{\frac{M}{2}}|\boldsymbol{\Sigma}_y|^{\frac{1}{2}}\exp\left(-\frac{1}{2}\sum_{j=1}^{L}\mathbf{y}_{\cdot j}^T\boldsymbol{\Sigma}_y^{-1}\mathbf{y}_{\cdot j}\right)\right) \\ &= L\log(|\boldsymbol{\Sigma}_y|)+\sum_{j=1}^{L}\mathbf{y}_{\cdot j}^T\boldsymbol{\Sigma}_y^{-1}\mathbf{y}_{\cdot j}. \end{aligned} \tag{5.6}$$

It is not expected that an explicit solution to the minimization problem can be found by differentiating and letting the expression equal to zero, hence it has to be solved iteratively based on a initial parameter guess $\boldsymbol{\gamma}^{(0)}$. One iterative method is the expectation maximisation (EM) algorithm. In general each iteration consist of an expectation (E) step, where a function determines the expectation of the likelihood function given the currently estimated parameters. The E-step is followed by an maximization (M) step which computes the parameters by maximizing the expected likelihood found in the E-step. In this case the E-step is to compute the posterior moments using (5.4) and (5.5) while the M-step is the following update rule of γ_i [23, p.147]

$$\gamma_i^{(k+1)} = \frac{1}{L}\|\boldsymbol{\mu}_{i\cdot}\|_2^2 + \boldsymbol{\Sigma}_{ii}, \quad \forall i = 1,\ldots,N.$$

The M-step is in general very slow on large data. An alternative is to use a fixed point update rule to fasten convergence on large data, however convergence is no longer ensured. The fixed point updating step is achieved by taking the derivative of the marginal log likelihood $\ell(\boldsymbol{\gamma})$ with respect to $\boldsymbol{\gamma}$ and equating it with zero. This lead to the following update rule which can replace the above M-step in the EM-algorithm

$$\gamma_i^{(k+1)} = \frac{\frac{1}{L}\|\boldsymbol{\mu}_{i\cdot}\|_2^2}{1-\gamma_i^{-1(k)}\boldsymbol{\Sigma}_{ii}}, \quad \forall i = 1,\ldots,N.$$

mangler at udlede denne regl

Empirically this alternative update rule have shown use full in highly under-determined large scale cases by driving many hyper parameters toward zero allowing for the corresponding weight in the source matrix to be discarded. For simultaneous sparse approximation problems this is the process referred to as multiple sparse Bayesian learning, M-SBL.

From the resulting $\boldsymbol{\gamma}^*$ the support set S of the source matrix $\mathbf{X}$ can be extracted,

$$S = \{i|\hat{\gamma}_i \neq 0\},$$

concluding the localization of active sources within $\mathbf{X}$. In practise some arbitrary small threshold can be used such that that any sufficiently small hyperparameter is discarded. For identification of the active sources the estimate of the source matrix $\mathbf{X}$ is given as $\mathbf{X}^* = \mathcal{M}^*$, with $\mathcal{M}^* = \mathbb{E}[\mathbf{X}|\mathbf{Y};\boldsymbol{\gamma}^*]$. This leads to the following estimate

$$\mathbf{X}^* = \begin{cases} \mathbf{x}_{i\cdot} = \boldsymbol{\mu}^*_{i\cdot}, & i \in S \\ \mathbf{x}_{i\cdot} = \mathbf{0}, & i \notin S \end{cases}$$

5.3.1 When k is Known

this section has to be written better

In this algorithm the number of active sources k is estimated as the non-zero entries of the hyper parameter $\boldsymbol{\gamma}$. However k has to be determined prior to the use of the M-SBL algorithm as it is used in the COV-DL algorithm. Thus there have been no reason to not pass the value of k on to this algorithm. By k being known in prior the tolerance determining when a parameter is close enough to zero are overruled. In stead the k largest hyper parameter are chosen to form S, which makes the support set of $\hat{\mathbf{X}}$.

5.3.2 Pseudo Code for the M-SBL Algorithm

Algorithm 2 M-SBL

procedure M-SBL($\mathbf{Y}, \mathbf{A}$, iterations)
  $\boldsymbol{\gamma} = \mathbf{1} \in \mathbb{R}^{\text{iterations}+2\times N\times 1}$
  iter = 0
  while $\boldsymbol{\gamma} \geq 10^{-16}$ **do**
    $\boldsymbol{\Gamma} = \text{diag}(\boldsymbol{\gamma}^{\text{iter}})$
    for $i = 1, \ldots, N$ **do**
      $\boldsymbol{\Sigma} = \boldsymbol{\Gamma} - \boldsymbol{\Gamma}\mathbf{A}^T\boldsymbol{\Sigma}_y^{-1}\mathbf{A}\boldsymbol{\Gamma}$
      $\mathcal{M} = \boldsymbol{\Gamma}\mathbf{A}^T\boldsymbol{\Sigma}_y^{-1}\mathbf{Y}$
      $\gamma_i^{(\text{iter}+1)} = \dfrac{\frac{1}{L}\|\boldsymbol{\mu}_{i\cdot}\|_2^2}{1 - \gamma_i^{-1(\text{iter})}\boldsymbol{\Sigma}_{ii}}$
    end for
    if iter = iterations **then**
      Break
    end if
    iter + = 1
  end while
  Return $\mathcal{M}^*, \boldsymbol{\gamma}^*$
end procedure
procedure SUPPORT($\mathcal{M}^*, \boldsymbol{\gamma}^*$, k)
  Support = $\mathbf{0} \in \mathbb{R}^k$
  $\boldsymbol{\gamma}_{\text{value}} = \boldsymbol{\gamma}^*(-2)$
  for j in range(k) **do**
    if $\boldsymbol{\gamma}_{\text{value}}(\arg\max(\boldsymbol{\gamma}_{\text{value}})) \mathrel{!}= 0$ **then**
      Support(j) = $\arg\max(\boldsymbol{\gamma}_{\text{value}})$
      $\boldsymbol{\gamma}_{\text{value}}(\arg\max(\boldsymbol{\gamma}_{\text{value}})) = 0$
    end if
  end for
  $\mathbf{X} = \mathbf{0} \in \mathbb{R}^{N\times L-2}$
  for i in Support **do**
    $\mathbf{X}(i) = \mathcal{M}^*(-1)(i)$
  end for
  Return $\mathbf{X}$
end procedure

5.3.3 Sufficient Conditions for Exact Source Localization

In [4] it is proven that exact source localization is guaranteed in the under-determined case, $k > M$ when the following conditions on $\mathbf{A}$ is fulfilled. The theorem is based on

a theoretical analysis of the minima where noise-free conditions are considered, that is letting $\sigma^2 \to 0$. Thus the following theorem applies to the noise less case. First, defined a function $f : \mathbb{R}^{M \times N} \to \mathbb{R}^{\frac{M(M+1)}{2} \times N}$, such that for $B = f(\mathbf{A})$ the j-th column is given as $\mathbf{b}_{\cdot j} = \text{vec}(\mathbf{a}_{\cdot j}\mathbf{a}_{\cdot j}^T)$. Here the function $\text{vec}(\cdot)$ corresponds to the function defined in section 4.1, being a vectorization of the lower triangular part of a matrix.

Theorem 5.3.1
Given a dictionary matrix $\mathbf{A}$ and a set of observed measurement $\mathbf{Y}$, M-SBL recovers the support set of any size k exactly in the noise-free case, if the following conditions are satisfied.

1. The active sources $\mathbf{X}_S$ are orthogonal. That is, $\mathbf{X}_S\mathbf{X}_S^T = \mathbf{\Lambda}$, where $\mathbf{\Lambda}$ is a diagonal matrix and S the support set.
2. $\text{Rank}(f(\mathbf{A})) = N$.

The proof can be found in [4, p. 16].

Chapter 6

Implementation and Verification

This chapter describes the implementation process of the main algorithm, where the two algorithms Cov-DL and M-SBL resulting from respectively chapter 4 and 5 are implemented and combined into one algorithm.

The implementation of each algorithm is initially tested on a simple simulated data set to verify the implementation. Next, both algorithms are tested on simulated data which aim to resemble real EEG measurements. By simulating the data set the true model parameters are known which allows for measuring the precision of the algorithms, based on a described error measurement. In addition different model variables are investigated in order to improve the model. Finally, the main algorithm is tested on the simulated data sets and a conclusion is made based on the results.

6.1 Implementation of Algorithms

In this section the implementation of the main algorithm is described with use of a flowchart which illustrate the flow through code. The main algorithm consist of three main stages, an initialization, Cov-DL for recovery of $\mathbf{A}$ and lastly M-SBL for recovery of $\mathbf{X}$. Considering figure 6.1, each stage of the algorithm is illustrated within one horizontal row of the flow diagram, furthermore the input and output are placed in their own row.

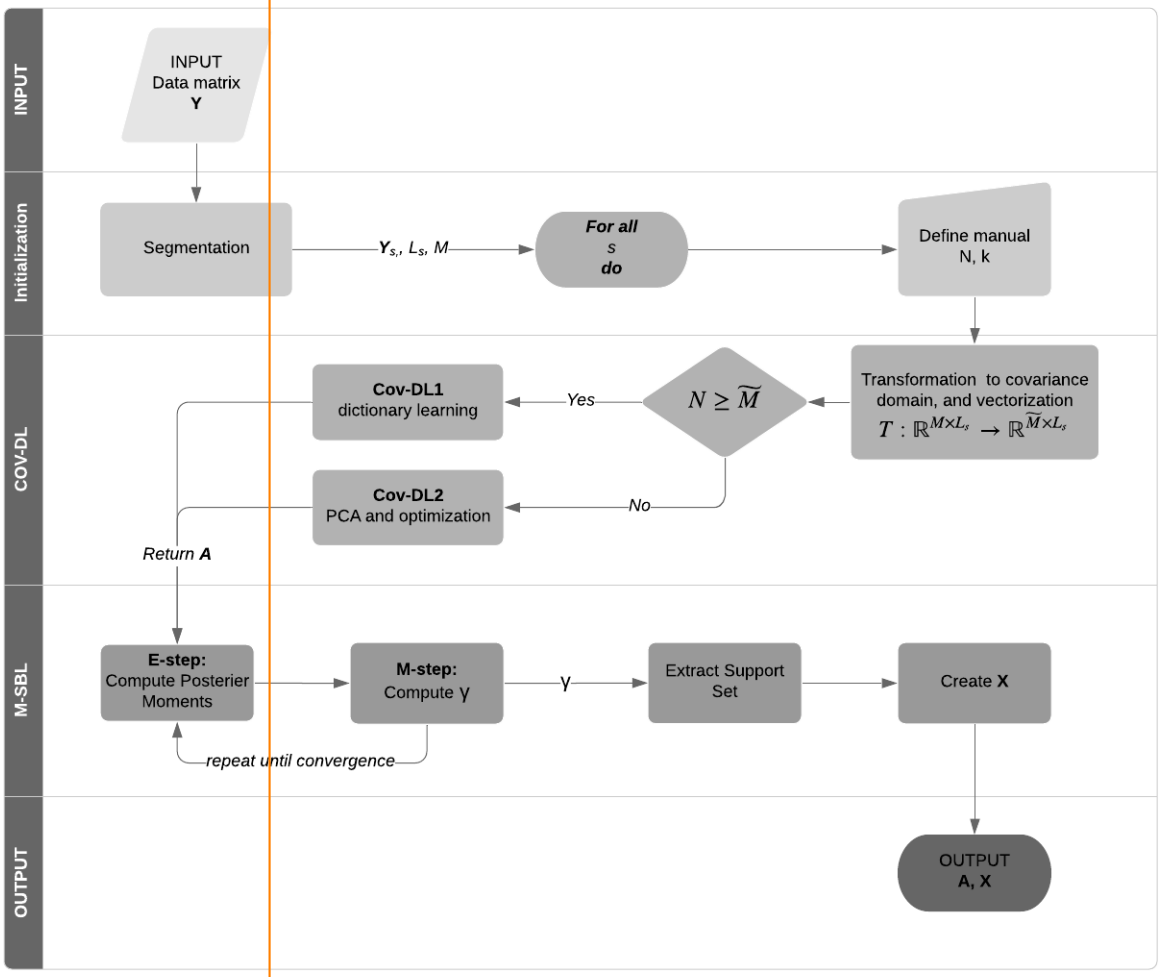

Figure 6.1: Flowchart illustrating the implementation of the main algorithm.

indsæt hat på gamma, X og A i flowchart og ændre data matrix Y til measurement matrix Y.

The input of the main algorithm consists of the measurement matrix $\mathbf{Y} \in \mathbb{R}^{M \times L}$, along with the corresponding sample frequency f. Within the initialization stage the measurement matrix $\mathbf{Y}$ go through a segmentation as described in subsection 3.2.1. The length of the segments are predefined by a time interval t in seconds such that $L_s = tf$. Each segment s is now specified by the measurement matrix $\mathbf{Y}_s \in \mathbb{R}^{M \times L_s}$. After the segmentation a loop is constructed such that for every segments s the remaining two stages, Cov-DL and M-SBL, of the main algorithm are performed. For each segment s, N and k are manually defined. This definition are either known in advance from the data or in the case of real EEG measurements they are unknown and a qualified guess must be made. With one segment and corresponding specifications of the expected number of active sources the second stage of the algorithm are initialised, recovery of $\mathbf{A}$. The implementation of Cov-DL stage follows algorithm 1

the manual input is an issue time wise when running over many segments, so the input should be made in advance

from section 4.3 closely thus only the main steps are illustrated on the flow diagram. First the measurement matrix $\mathbf{Y}_s$ for all segments are transformed to the covariance domain and then vectorized, resulting in an extension of the dimensionality from M to $\widetilde{M} = \frac{M(M+1)}{2}$. Next, the estimation of $\mathbf{A}$ is performed from either Cov-DL1 or Cov-DL2 depending on the relation between $\widetilde{M}$ and N, as described respectively in section 4.2.1 and 4.2.2. The estimate $\hat{\mathbf{A}}$ serves as the input to the next stage, M-SBL for recovering of $\mathbf{X}$, along with the measurement matrix $\mathbf{Y}_s$. The last stage of the main algorithm consists of the iterative EM algorithm for maximizing the marginal likelihood (5.6) with respect to $\boldsymbol{\gamma}$, which is the hyperparameter from which $\mathbf{X}$ can be determined as described in section 5.3. Lastly, the output of the main algorithm $\hat{\mathbf{X}}$ and $\hat{\mathbf{A}}$ is illustrated on the flowchart.

should we define new number of "samples" as well?

6.1.1 Coding Practice

The implementation of the main algorithm in performed in Python 3.6. The software and guide to run the scripts are available through appendix **??**.

The practical implementation process are based on module development. The established model and the three stages of the main algorithm makes a superior system design. For each stage the necessary tasks are identified and divided into smaller modules. For each module the task is specified and the algorithms are established and implemented. This is followed by a test of the module and possible modifications until the task is performed without error. Due to the time limitation of this project the software was developed along side the dynamic research process. Thus the specifications to some modules have been redefined and hence the modification process are repeated. Finally, the modules are united into one stage for which tests are performed, and lastly all the stages are united to the resulting main algorithm.

The software are based on functions, for example one module is specified by one function, for which docstrings is used, following NumPy docstring format[1] allowing insight into the structure and thoughts behind the different software elements.

For each of the stages, Cov-DL and M-SBL, verification and performance tests will be performed and described later in this chapter, followed by testing phase of the main algorithm.

6.2 Data Simulation

To test the performance of the algorithm simulated data, corresponding to the model $\mathbf{Y} = \mathbf{AX}$, is needed. All data sets are simulated based on the following approach, satisfying the sufficient conditions for recovery, displayed in theorem 5.3.1.

A source matrix $\mathbf{X} \in \mathbb{R}^{N \times L}$ is constructed, such that each row makes an independent signal which varies over L samples or a zero row. As such the non-zero

[1] `https://numpydoc.readthedocs.io/en/latest/`

rows of $\mathbf{X}$ are mutually orthogonal, which fulfils the first conditions of theorem 5.3.1. Then a mixing matrix $\mathbf{A} \in \mathbb{R}^{M \times N}$ is constructed with equally distributed and independent entries. As such the source signals are randomly mixed and the mixing matrix fulfils the second condition of theorem 5.3.1. With known $\mathbf{A}$ and $\mathbf{X}$ the measurement matrix $\mathbf{Y} \in \mathbb{R}^{M \times L}$ is simulated according to the model, by the matrix product $\mathbf{Y} = \mathbf{AX}$.

Two different kinds of data sets are simulated. One deterministic data set with simple and predictable source signals to ensure a solution and easy visualization. And a stochastic data set with randomized and fluctuating source signals to resemble realistic EEG measurements.

6.2.1 Deterministic Data Set

Two different deterministic data sets are simulated, with a different number of zero rows. One specified by $N = 5$, $k = 4$, $M = 3$ and $L = 1000$. That is a source matrix $\mathbf{X}$ with 4 independent signals and 1 zero row which is mixed into a measurement matrix with 3 measurement per sample. The second deterministic data set is specified by $N = 8$, $k = 4$, $M = 3$ and $L = 1000$. This is 3 additional zero rows. From the specifications the first data set comply to $N \leq \frac{M(M+1)}{2}$ which imply the use of Cov-DL2. The second data set comply to $N > \frac{M(M+1)}{2}$ and $k \leq \frac{M(M+1)}{2}$ implying the use of Cov-DL1. As such it is possible to test both branches of the Cov-DL algorithm.

The 4 independent source signals of $\mathbf{X}$ are defined by

1. a sinus signal $\sin(2t)$
2. a sawtooth signal with period $2\pi t$
3. a sinus signal $\sin(4t)$
4. a sign function of a sinus signal $\sin(3t)$

with t being a time index defined in the interval $[0, 4]$ with L samples. Each of the four signals are randomly drawn and used to construct a source matrix $\mathbf{X}$ of size $k \times L$, then zero rows are inserted randomly, such that $\mathbf{X} \in \mathbb{R}^{N \times L}$. The mixing matrix $\mathbf{A}$ of size $M \times N$ is randomly generated from a Gaussian distribution. By multiplying the source matrix and the mixing matrix the measurement matrix $\mathbf{Y}$ is achieved. The deterministic data set then consist of $\{\mathbf{Y}, \mathbf{X}, \mathbf{A}\}$.

In figure 6.2 the first deterministic data set is illustrated by the source signals plotted in the top and the measurement signals plotted in the bottom. This illustrates how the source signals are transformed by the mixing matrix $\mathbf{A}$.

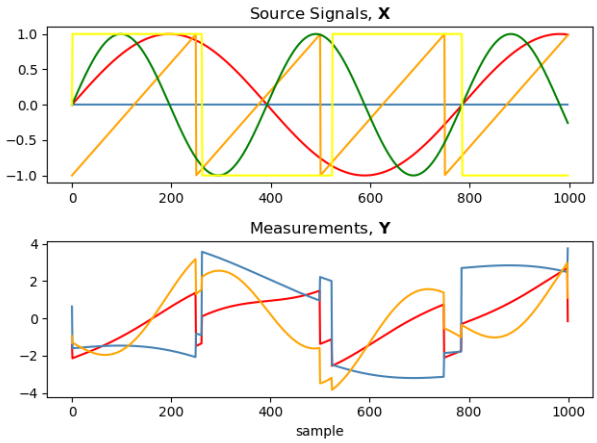

Figure 6.2: Visualization of the signals of the source matrix $\mathbf{X}$ in comparison to the measurement signals of the measurement matrix $\mathbf{Y}$ from the deterministic data set specified by $N = 5, M = 3$, $k = 4$ and $L = 1000$.

6.2.2 Stochastic Data Set

The purpose of this second kind of data is to resemble EEG measurements for which the model is intended. Here different data sets are simulated depending on the chosen specifications of N, k, M and L. Every data set is constructed based on four different autoregressive processes of various order, each process representing one source signal

- $x_t^1 = \sum_{i=1}^{2} \phi_i x_{t-i}^1 + w_t^1$
- $x_t^2 = \sum_{i=1}^{2} \zeta_i x_{t-i}^2 + w_t^2$
- $x_t^3 = \sum_{i=1}^{3} \eta_i x_{t-i}^3 + w_t^3$
- $x_t^4 = \sum_{i=1}^{4} \xi_i x_{t-i}^4 + w_t^4$

where $\boldsymbol{\phi}, \boldsymbol{\zeta}, \boldsymbol{\eta}$ and $\boldsymbol{\xi}$ are different model parameters and w_t^j for $j = 1, \cdots, 4$ is white noise, corresponding to process j. $\mathbf{X}$ is constructed by drawing k autoregressive processes randomly among the four, if $k < N$ zero rows are inserted randomly such that $\mathbf{X} \in \mathbb{R}^{N \times L}$. The mixing matrix $\mathbf{A}$ of size $M \times N$ is, like the previously, generated randomly from a Gaussian distribution. By multiplying the source matrix and the mixing matrix, the measurement matrix $\mathbf{Y}$ is achieved. The stochastic data set then consist of $\{\mathbf{Y}, \mathbf{X}, \mathbf{A}\}$.

One simulation of a stochastic data set is illustrated in figure 6.3. The illustrated data set is specified by $N = 5$, $M = 3$, $k = 4$ and $L = 1000$.

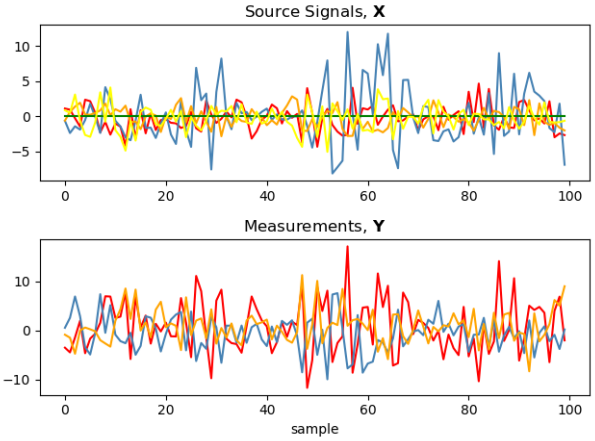

Figure 6.3: Visualization of the source signals of the source matrix $\mathbf{X}$ in comparison to the measurement signals of the measurement matrix $\mathbf{Y}$ from a stochastic data set specified by $N = 5$, $M = 3$, $k = 4$ and $L = 1000$. For simplicity only samples [0:100] are plotted.

6.2.3 Error Measurement

To evaluate performance of the algorithms it is evident to look at the differences between the true and estimated matrices, mixing matrix $\mathbf{A}$ and source matrix $\mathbf{X}$ – which is possible due to the input data being simulated. For this task the mean squared error (MSE) has been chosen. The MSE measures the average squared difference between some estimated value and the true value. For $\hat{\mathbf{g}}$ being the estimate of the vector $\mathbf{g}$ the MSE can be written as

$$\mathrm{MSE}(\mathbf{g}, \hat{\mathbf{g}}) = \frac{1}{T}\sum_{i=1}^{T}(g_i - \hat{g}_i)^2,$$

with T being the number of elements in the vector $\mathbf{g}$.

For this project the estimates form a matrix. Here the MSE is computed for each row, which for $\mathbf{X}$ is the estimate of one source signal, then the resulting MSE is the average over all rows. For $\mathbf{X}, \hat{\mathbf{X}} \in \mathbb{R}^{N\times L}$ the MSE is written as

$$\mathrm{MSE}(\mathbf{X}, \hat{\mathbf{X}}) = \frac{1}{N}\sum_{i}^{N}\left(\frac{1}{L}\sum_{j=1}^{L}(\mathbf{X}_{ij} - \hat{\mathbf{X}}_{ij})^2\right).$$

Similarly, the MSE can be written for $\mathbf{A}, \hat{\mathbf{A}} \in \mathbb{R}^{M\times N}$.

The MSE is viewed as a measure of the quality of an estimator, in this case of how M-SBL and Cov-DL perform. For a large MSE the estimated values are dispersed widely around its mean while for a small MSE value the estimated matrix/values is closely dispersed around the mean. Usually, a small MSE value indicates a good estimator but the value cannot be to small as this would indicate that the data has

lige finde en god
m siger dette

been overfitted . Therefore, a good MSE and therefore a good performance would be depending on how the data is scattered as widely scattered data may lead to a MSE value not close to zero but it would still be the a good measure for the estimator.

MSE:måske dette kan uddybes lidt i forhold til hvordan vores målingere opføre sig, ved vi hvad vi kan forvente

6.3 Verification of Algorithms

In this section the implementation of Cov-DL and M-SBL are verified separately based on the MSE between the true- and the estimated model parameters. During the tests on simulated data sets the segmentation stage is ignored by letting the simulated data form one single segment.

6.3.1 Test of Cov-DL

As seen from the flowchart 6.1 Cov-DL takes a measurement matrix $\mathbf{Y}$, N and k as input and return an estimation $\hat{\mathbf{A}}$ of the mixing matrix $\mathbf{A}$. The Cov-DL algorithm is tested on the two simulations of the deterministic data, specified in section 6.2.1.

Cov-DL1

For $\mathbf{Y}$ specified by $N > \widetilde{M}$ and $k \leq \widetilde{M}$, implying Cov-DL1, the true and estimated values of $\mathbf{A}$ are plotted in figure 6.4 for visual comparison. Note that each matrix is vectorized such that the corresponding entries are compared. The resulting $\text{MSE}(\mathbf{A}, \hat{\mathbf{A}})$ between the true $\mathbf{A}$ and the estimated $\hat{\mathbf{A}}$ become

$$\text{MSE}(\mathbf{A}, \hat{\mathbf{A}}) = 1.37.$$

From figure 6.4 it is seen that the precision of the estimate varies significant for each entry. Though, values within the a similar range are obtained. Furthermore, the $\text{MSE}(\mathbf{A}, \hat{\mathbf{A}})$ is fairly small suggesting that the estimate is acceptable

how should we determined this?? and can we give an better explanation?.

Cov-DL2

For $\mathbf{Y}$ specified by $N \leq \widetilde{M}$, implying Cov-DL2, the true and estimated values of $\mathbf{A}$ are plotted in figure 6.5 for visual comparison. Additionally, is the initial $\mathbf{A}_{\text{ini}}$, which is given to the optimization solver within Cov-DL2, plotted in the same figure, as standard that is a Gaussian matrix. The resulting MSE between the true $\mathbf{A}$ and the estimated $\hat{\mathbf{A}}$ become

$$\text{MSE}(\mathbf{A}, \hat{\mathbf{A}}) = 2.46.$$

From figure 6.5 the estimate $\hat{\mathbf{A}}$ shows visual tendencies from the true $\mathbf{A}$. However, when it is compared to the initial guess of $\mathbf{A}$, $\mathbf{A}_{\text{ini}}$, it is seen that the estimate $\hat{\mathbf{A}}$ have moved further away from the true $\mathbf{A}$ compared to $\mathbf{A}_{\text{ini}}$. This suggest some flaw within the optimization process. By printing the convergence message from the used optimization solver it is confirmed that the optimization process was found to be terminated successful, with a current cost function value at 0.0 after 26 iterations. This suggest that an global minima has been found, but the minima, $\hat{\mathbf{A}}$, do not correspond to the true $\mathbf{A}$. To confirm this the following evaluations of the cost function was conducted.

$$\text{cost}(\hat{\mathbf{A}}) = 0.0$$
$$\text{cost}(\mathbf{A}_{\text{ini}}) = 1.564$$
$$\text{cost}(\mathbf{A}) = 1.809$$

These evaluations ensures that the optimization solver have manage to find the solution that minimizes the cost function. By evaluating the cost function with respect to the true $\mathbf{A}$ is it seen that it is not a global minimizer to the optimization problem. This suggest that the optimization problem, derived in section 4.2.2, do not fulfil the purpose.

better word to use here?

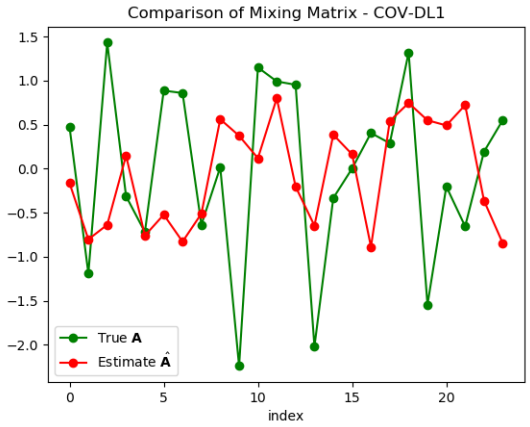

Figure 6.4: Estimated values of $\hat{\mathbf{A}}$ compared to the true values $\mathbf{A}$

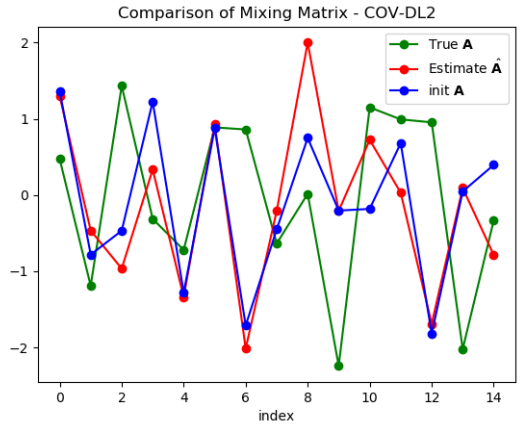

Figure 6.5: The initial $\mathbf{A}_{\text{ini}}$ and the estimate $\hat{\mathbf{A}}$ compared to the true values $\mathbf{A}$.

Conclusion to Estimate $\hat{\mathbf{A}}$

From the results of the evaluation of cost function it is found that the estimate $\hat{\mathbf{A}}$, especially within the Cov-DL2 branch, can not be consider a valid estimate of the mixing matrix $\mathbf{A}$. It is suggested that the flaw lies within either the derivation of the cost function to the optimization problem, more specifically within the assumption made throughout the derivation concerning the relation between $\mathbf{A}, \mathbf{D}$ and $\mathbf{U}$. This

statement build upon the success of the (non-documented) unit tests of the Cov-DL2 algorithm suggesting that the optimization of $\mathbf{D}$ not depending on $\mathbf{A}$ is possible. The appearance of this issue may suggest that there is a lack within the published results [3]considering the possibility of reconstructing the results.

(tjek cite)

Due to the time limitation of the project the error is not investigated further, and it is concluded that the estimate of $\mathbf{A}$ is not valid hence it will not be used as an input for the next stage of the main algorithm, M-SBL. This conclusion suggest that an alternative action must be considered. This is discussed further in section 6.4.

6.3.2 Test of M-SBL

From the flowchart 6.1 it seen that that the M-SBL algorithm takes $\hat{\mathbf{A}}$ and $\mathbf{Y}$ as input. The algorithm is now tested on the same two deterministic data sets specified by $M = 3$, $k = 4$, $L = 1000$ and respectively $N = 5$ and $N = 8$ as used above. In order to not let the performance of Cov-DL affect the result of M-SBL the true mixing matrix $\mathbf{A}$ is used as input along with the corresponding $\mathbf{Y}$.

The estimate $\hat{\mathbf{X}}$ is plotted in figure 6.6 and 6.7. The source signals equal to zeros of the estimate $\hat{\mathbf{X}}$ is not plotted so the figures do not visualize the exact localization of the source signals.

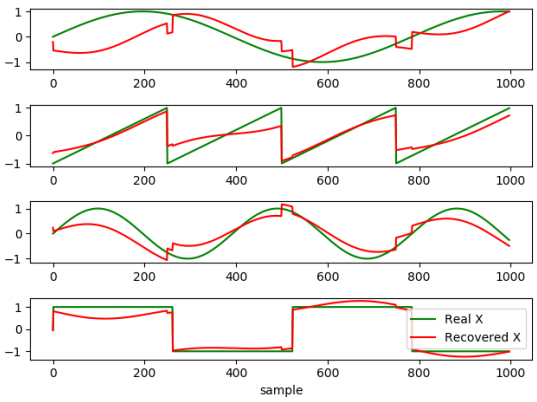

Figure 6.6: Estimated values of $\hat{\mathbf{X}}$ compared to the true values $\mathbf{X}$. From measurement $\mathbf{Y}$ specified by $N = 5$, $M = 3$, $k = 4$ and $L = 1000$ and the true mixing matrix $\mathbf{A}$.

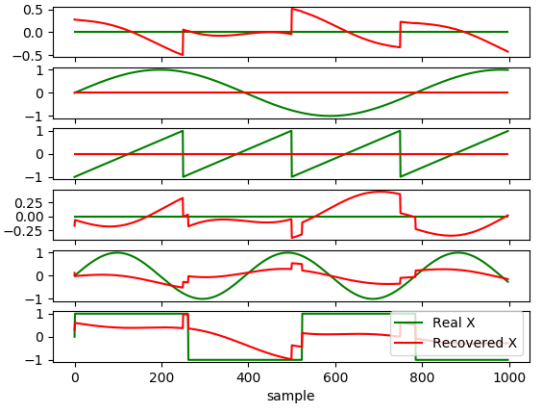

Figure 6.7: Estimated values of $\hat{\mathbf{X}}$ compared to the true values $\mathbf{X}$. From measurement $\mathbf{Y}$ specified by $N = 8$, $M = 3$, $k = 4$ and $L = 1000$ and the true mixing matrix $\mathbf{A}$.

The resulting MSE between the true $\mathbf{X}$ and the estimate $\hat{\mathbf{X}}$ from figure 6.6 with $N = 5$, becomes

$$\text{MSE}(\mathbf{X}, \hat{\mathbf{X}}) = 0.127.$$

From figure 6.6 it is seen that all four source signals are recovered at the right locations. As suggested by the achieved MSE the estimate are not exact, but it is clear that the estimates, by looking at figure 6.6, manage to follow the right pattern of the true signals.

The resulting MSE between the true $\mathbf{X}$ and the estimated $\hat{\mathbf{X}}$ from figure 6.7 with $N = 8$ thus more sparse, become

$$\text{MSE}(\mathbf{X}, \hat{\mathbf{X}}) = 0.228.$$

From figure 6.7 it is seen that most of the source signals are recovered but dislocated. This indicate that the algorithm can manage to locate the source signal however the more options the greater chance of dislocation.

Possibilities of $N = k$

From the problemstatement in chapter 2 it is an issue that k has to be known in prior, in order to estimate $\mathbf{A}$ and $\mathbf{X}$. A short discussion in subsection 5.3.1, describes how k can be estimated within the M-SBL algorithm. However one still needs to provide k in order to estimate $\mathbf{A}$, thus an qualified estimate of k can not be avoided.

Similar to k, the maximum number of active sources N is unknown in practise as it is described in chapter 1. The difference between k and N defines the number of zero rows in $\mathbf{X}$. During the estimation of $\mathbf{X}$ the localisation of the non-zero rows are in general significant to minimize the MSE. However, the fact that the true N can not be known for EEG measurements weakens the argument for of focusing on the localisation rather than only focusing on the value estimation of the source signals. Furthermore it is not within the main scope of this thesis to localise the source signal. When considering the linear system, $\mathbf{Y} = \mathbf{AX}$, of which the model is build, $\mathbf{Y}$ does not change by removing the zero-rows of $\mathbf{X}$ and the corresponding columns in $\mathbf{A}$.

From this is can be argued that letting $N = k$ is as good an estimate of N as any other. Note in order to fulfil the sufficient conditions for the existence of a solution to the system, that k hence N is limited by $\widetilde{M}$.

Consider the effect of letting $N = k$ within the M-SBL algorithm. Here it is only the estimation of the support set which is eliminated.

Figure 6.8 show the estimated sources signal for a simulation of the deterministic data set now specified by $N = k = 4$ $M = 3$ and $L = 1000$. The resulting MSE become

$$\mathbf{X}_{MSE} = 0.121$$

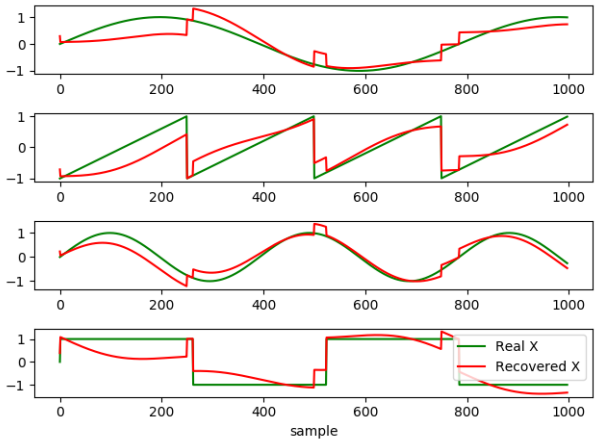

Figure 6.8: Estimated values of $\hat{\mathbf{X}}$ compared to the true values $\mathbf{X}$. From deterministic data $\mathbf{Y}$ specified by $N = k = 4$, $M = 3$ and $L = 1000$

From the above discussion and the results in figure 6.8 it is confirmed that letting $N = k$ has no disadvantage with respect to the results, when the localisation of the source signal is not a priority. Thus is it chosen that $N = k$ will be used throughout the thesis.

6.3.3 Test on Stochastic Data Sets

The M-SBL algorithm is now tested on two stochastic data sets which resembles the real EEG measurements. The first stochastic data set is simulated with specification $N = k = 8$, $M = 6$, $L = 1000$. The resulting estimate is plotted in figure 6.9 and the MSE becomes

$$\text{MSE}(\mathbf{X}, \hat{\mathbf{X}}) = 1.643.$$

The second stochastic data set is simulated with specification $N = k = 16$, $M = 6$, $L = 1000$. This tests the capabilities of the M-SBL algorithm when the distance between M and N is enlarged. The performance relative to the relation between N and M is further investigated for the main algorithm in section 6.4. The resulting estimate is plotted in figure 6.10 and the MSE becomes

$$\text{MSE}(\mathbf{X}, \hat{\mathbf{X}}) = 5.182.$$

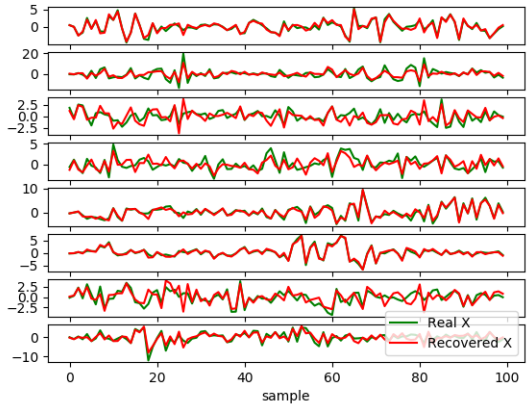

Figure 6.9: Estimated values of $\hat{\mathbf{X}}$ compared to the true values $\mathbf{X}$. From measurement $\mathbf{Y}$ specified by $N = k = 8$, $M = 6$ and $L = 1000$ and the true mixing matrix $\mathbf{A}$

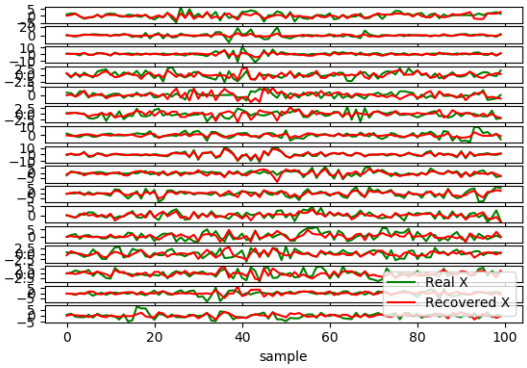

Figure 6.10: Estimated values of $\hat{\mathbf{X}}$ compared to the true values $\mathbf{X}$. From measurement $\mathbf{Y}$ specified by $N = k = 16$ $M = 6$ and $L = 1000$ and the true mixing matrix $\mathbf{A}$

From figure 6.9 it is visually confirmed that the M-SBL algorithm manage to recover the source signals of the source matrix $\mathbf{X}$. Some source signals nearly perfect estimated while other are have minor differences. From figure 6.10 the same tendency is seen, though more visual flaws are seen compared to figure 6.9. This result suggest that a bigger distance between M and N results in a weaker performance from the M-SBL algorithm.

6.4 Test of the Main Algorithm

In this section the performance of the main algorithm is tested.

6.4.1 Alternative to Estimate $\hat{\mathbf{A}}$

As concluded the Cov-DL algorithm do not recover a sufficient estimate of the mixing matrix $\mathbf{A}$, therefore a different approach is necessary.

Replacing the insufficient estimate by a fixed estimate $\hat{\mathbf{A}}_{\text{fix}}$ is one immediately solution. This choice is supported by the observations from Cov-DL2 where $\mathbf{A}_{\text{ini}}$ matrix provides an estimate which is happens to be a least as good as the one provided by Cov-DL. Thus the challenge is now to determine a fixed matrix for which its characteristics resembles those of the true mixing matrix. However, from chapter 1 it is clear that no specific characteristic of the mixing matrix is known, which supports the choice of an random matrix of Gaussian distribution or similar, as it was chosen for the initial guess $\mathbf{A}_{\text{ini}}$ for the estimate. From this perspective three fixed mixing matrices are defined, by drawing each entry from a specified distribution:

$$\hat{\mathbf{A}}_{\text{uni}} \sim \mathcal{U}(-1, 1)$$

$$\hat{\mathbf{A}}_{\text{norm}} \sim \mathcal{N}(0, 2)$$

$$\hat{\mathbf{A}}_{\text{gauss}} \sim \mathcal{N}(0, 1)$$

Note that the third matrix $\hat{\mathbf{A}}_{\text{gauss}}$ is generated the same way as the true mixing matrix of the stochastic data sets. Thus it is expected to have the lowest MSE when compared to the true mixing matrix $\mathbf{A}$. However, it is of interest to investigate whether it is the best estimate of $\mathbf{A}$ which provide the best estimate of $\mathbf{X}$.

A different option regarding a choice for a fixed $\hat{\mathbf{A}}$ is to utilize the ICA algorithm, described in appendix **??**. By the ICA algorithm it is possible to solve the EEG inverse problem for both $\mathbf{A}$ and $\mathbf{X}$, in the case where $k \leq M$. Consider a simulation of a stochastic data set specified by $N = k = M$. Solving the system by ICA yields an estimate of $\mathbf{A}$. Now reduce the data set $\mathbf{Y}$ such that $M \leq k$. Similar the estimate of $\mathbf{A}$ is reduced by removing the same rows as in $\mathbf{Y}$, this yields the an estimate $\hat{\mathbf{A}}_{\text{ICA}}$ which can be used as a fixed input to M-SBL along with the corresponding reduced $\mathbf{Y}$.

The four different fixed estimates $\hat{\mathbf{A}}$ are tested on stochastic data sets specified by $M = 10$, $N = k = 16$ and $L = 1000$, where the estimate $\hat{\mathbf{A}}_{\text{ICA}}$ has been reduced to $M = 10$. To get an average performance 50 different simulations are conducted with the same specifications, each system $\mathbf{X}$ is estimated from each of the four fixed estimates of $\mathbf{A}$[2], and the MSE are computed. The resulting averaged $\text{MSE}(\mathbf{A}, \hat{\mathbf{A}}_{\text{fix}})$ and $\text{MSE}(\mathbf{X}, \hat{\mathbf{X}})$ are visualised in figure 6.11, for each of the four $\hat{\mathbf{A}}_{\text{fix}}$. Furthermore, the plotted values are found in table 6.1.

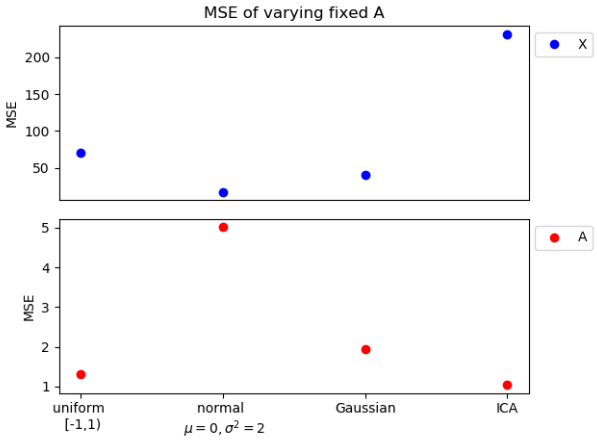

Figure 6.11: Average MSE values for each of the four fixed mixing matrix $\hat{\mathbf{A}}_{\text{fix}}$ resulting from a stochastic data set specified by $M = 10$, $N = k = 16$ and $L = 1000$.

[2]Note that for each of the 50 repetitions four new $\hat{\mathbf{A}}_{\text{fix}}$ are fixed.

	$\hat{\mathbf{A}}_{\text{uni}}$	$\hat{\mathbf{A}}_{\text{norm}}$	$\hat{\mathbf{A}}_{\text{gauss}}$	$\hat{\mathbf{A}}_{\text{ICA}}$
$\text{MSE}(\mathbf{A}, \hat{\mathbf{A}}_{\text{fix}})$	1.314	5.021	1.935	1.033
$\text{MSE}(\mathbf{X}, \hat{\mathbf{X}})$	70.74	17.36	40.58	231.4

Table 6.1: Average MSE values resulting from stochastic data set specified by $M = 10$, $N = k = 16$ and $L = 1000$ with a fixed estimate of the mixing matrix $\hat{\mathbf{A}}_{\text{fix}}$.

From table 6.1 and figure 6.11 it is first of all seen that relation between the MSE of $\mathbf{A}$ and $\mathbf{X}$ is not as expected, as the lowest $\text{MSE}(\mathbf{A}, \hat{\mathbf{A}}_{\text{fix}})$ results in the highest $\text{MSE}(\mathbf{X}, \hat{\mathbf{X}})$ and so forth. The lowest $\text{MSE}(\mathbf{A}, \hat{\mathbf{A}}_{\text{fix}})$ is achieved by using $\hat{\mathbf{A}}_{\text{ICA}}$, which confirms that the ICA algorithm manage to estimate $\mathbf{A}$ when $k \leq M$. However, as this do not result in the best estimate of $\mathbf{X}$ a different choice of $\hat{\mathbf{A}}$ is still considered. The lowest $\text{MSE}(\mathbf{X}, \hat{\mathbf{X}})$ is achieved by use of $\hat{\mathbf{A}}_{\text{norm}}$, which resulted in the largest $\text{MSE}(\mathbf{A}, \hat{\mathbf{A}}_{\text{fix}})$.

As the main interest in this thesis is to identify and localize the active sources of EEG measurements a low $\text{MSE}(\mathbf{X}, \hat{\mathbf{X}})$ is more desirable than a low $\text{MSE}(\mathbf{A}, \hat{\mathbf{A}}_{\text{fix}})$. Furthermore, a disadvantage of using $\hat{\mathbf{A}}_{\text{ICA}}$ is the limitations in practice when $k = M$ is not possible. From these observation a fixed estimate of the mixing matrix drawn from a normal distribution with mean 0 and variance 2, $\hat{\mathbf{A}}_{\text{norm}}$, is chosen as the the alternative estimate of $\mathbf{A}$, and is used throughout the thesis.

6.4.2 Performance Test of Main Algorithm

In order to evaluate the performance of the main algorithm tests are conducted on several simulated stochastic data sets with different specification. The aim is to see how the relationship between N and M affect the performance, in other words how robust the algorithm is towards low density measurements. The main algorithm is tested on simulated stochastic data sets specified by $M = 8$, $L = 1000$, $k = N$ with N in the range $N = [M + 1, \ldots, 36]$, as such $k < \widetilde{M}$ is withhold insuring a solution. For each value of N ten different data sets are simulated and solved, and the average $\text{MSE}(\mathbf{X}, \hat{\mathbf{X}})$ are used as the result. The average $\text{MSE}(\mathbf{X}, \hat{\mathbf{X}})$ are plotted in figure 6.12.

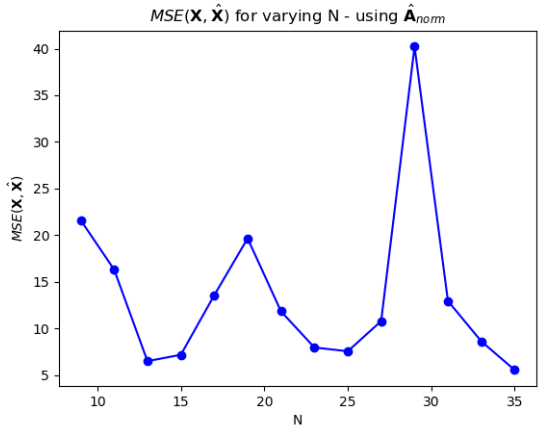

Figure 6.12: Visualization of $\text{MSE}(\mathbf{X}, \hat{\mathbf{X}})$ of the main algorithm with simulated stochastic data sets specified by $M = 8$, $L = 1000$ and $k = N$ for $N = M + 1, \cdots, 36$. Average over 10 repetitions for each N.

From figure 6.12 it is seen that the $\text{MSE}(\mathbf{X}, \hat{\mathbf{X}})$ lies in the interval $[4, 14]$, however no clear trend appears in the plot. This suggest that it is not an representative average which have been plotted, thus the test is repeated with more repetitions for value of N. The new result of the $\text{MSE}(\mathbf{X}, \hat{\mathbf{X}})$ is seen in figure 6.13.

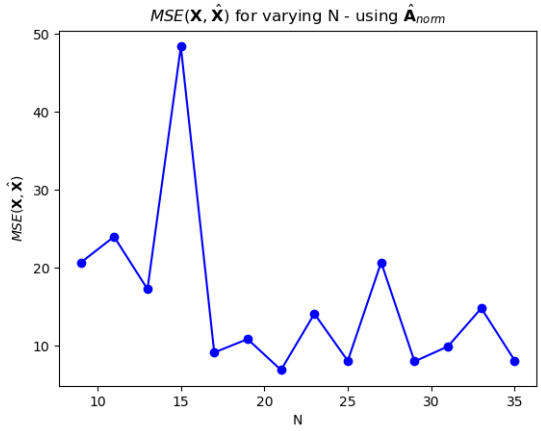

Figure 6.13: Visualization of $\text{MSE}(\mathbf{X}, \hat{\mathbf{X}})$ of the main algorithm with simulated stochastic data sets specified by $M = 8$, $L = 1000$ and $k = N$ for $N = M + 1, \cdots, 36$. Average over 500 repetitions for each N.

Figure 6.13 confirms the result of the first test. Thus it must be that average behaviour which is seen. This suggest that the performance of the main algorithm is not affected by the relation between M and N. However, this assumption is counter intuitive and it is a contradiction to the results seen in figure 6.9 and 6.10, where

the true $\mathbf{A}$ was utilised. Thus the choice of the alternative estimate $\hat{\mathbf{A}}_{\text{norm}}$ might have influenced the results negatively. Furthermore it is worth to notice the relative large interval of the $\text{MSE}(\mathbf{X}, \hat{\mathbf{X}})$ suggesting a vary high variance within the resulting $\text{MSE}(\mathbf{X}, \hat{\mathbf{X}})$, which add a certain unreliability to the results.

6.5 Conclusion

Through this chapter the implementation process has been described, followed by verification tests of the two main stages of the main algorithm, respectively the Cov-DL algorithm and the M-SBL algorithm.

From the test of M-SBL on stochastic data sets it was verified that the algorithm provide the expected output, and from $\text{MSE}(\mathbf{X}, \hat{\mathbf{X}})$ and the corresponding visual comparison, the estimate was found to be sufficient. The verification of M-SBL was conditioned on the true mixing matrix $\mathbf{A}$ as input, to not let the precision of the estimate $\hat{\mathbf{A}}$ from Cov-DL affect the results. Furthermore, the possibilities of letting $k = N$ was discussed. Either N nor k is known in practise, but one has to provide the best guess for both N and k to the algorithm in order to provide corresponding number of source signals. By letting $k = N$ one only has to guess the maximal number of active sources and not the relation between active and non-active sources, which is considered easier. Considering the consequences within the M-SBL, letting $k = N$ will reduce the chance of dislocation, which is seen as an advantage. Furthermore, tests on the deterministic data sets confirmed that the estimated active sources was not degraded. Thus it is confirmed that letting $k = N$ is sufficient, and will be used when testing the main algorithm on real EEG measurements.

From the verification test of Cov-DL, providing the estimate $\hat{\mathbf{A}}$, it was found that Cov-DL did not manage to provide a sufficient result. Is was confirmed that the Cov-DL resulted in the expected output relative to the implementation, but the output did not comply with the theoretically expected result. Thus it is concluded that the theory provided by [5] was misinterpreted, suggesting partly that the degree of reproducibility of the paper have not been sufficient. Due to the time scope of the thesis this issue is not investigated further. However, as the estimate of $\mathbf{A}$ resulting from Cov-DL is crucial in order to estimate the source signals from real EEG measurements, it was chosen that the best possible alternative to original estimate must be used, in order to pursue the remaining elements of the thesis. Then, the missing estimate must be taking into account when evaluating the final results.

Different suggestions for an alternative estimate of $\mathbf{A}$ was proposed an evaluated by the resulting $\text{MSE}(\mathbf{X}, \hat{\mathbf{X}})$. Here it was found that the fixed estimate $\hat{\mathbf{A}}_{\text{norm}}$ generated from a normal distribution with mean 0 and variance 2 provided the best result, when tested on stochastic data sets resembling real EEG measurements.

Lastly the performance of the main algorithm was tested on stochastic data sets. Here tests were performed on varying N in order to investigate performance

relative to the relation between M and N. For each value of N repetitions was conducted and the average MSE$(\mathbf{X}, \hat{\mathbf{X}})$ was evaluated. The MSE$(\mathbf{X}, \hat{\mathbf{X}})$ was found to lie within an interval from 2 to 25, without any characteristic trend relative to the increasing N. From this is it concluded that the performance do not rely on the relation between N and M. Despite that this was indicated by the tests where the true $\mathbf{A}$ was utilised. Thus the lack of a precise estimate of $\mathbf{A}$ might influence the final results.

Overall, the implementation of the main algorithm is approved. However the performance is not as good as expected. From this the main algorithm is ready to be tested on real EEG measurements in order to evaluated the performance with respect to the problem statement of this thesis. These tests are specified and conducted in the next chapter.

Chapter 7

Test on EEG measurements

Through this chapter a modification of the main algorithm will be described such that the main algorithm can handled real EEG measurements.

For verification of the results the MSE values achieved from the EEG measurements will be compared to MSE values achieved from the use of ICA, cf. appendix. This would include some preprocessing of the used EEG measurements.

At last, from the observed results of the comparison a conclusion will close this chapter.

7.1 Data Description

The data set of interest in this chapter consist of real EEG scalp measurements which has been provided for this thesis. The EEG scalp measurements are achieved from a experiment with a EEG cap where the test person was closing and opening his/her eyes. For the experiment a cap with $M = 27$ sensors has been used to measure the mixing of sources over a period of XX seconds. The data set then only consist of the measurement matrix $\mathbf{Y} \in \mathbb{R}$ and not the source matrix and mixing matrix as the data sets described in chapter 6.

- S1_Cclean is clean data for the forst subject for closed-yeys condition
- 27 channels with names and position in EEG.chanlocs structure
- Preprocessed: The data is bandpass filtered between 1 and 40 Hz. Then decomposed by ICA and the independent components related to eye activity was removed.
- S1_Cclean is divided into 144 segments (0ne second long), with 27 sensor and 515 samples each (144 x 27 x 515). X is found from ICA and is of size 144 x 27 x 513. X_nonzero is X from ICA consisting of only active sources (144 x k x

513) where k is different for each segment (k is 144 long). The nonzero values if found from a tolerance of 10E-03 and -10E-03 such that at box around zero is equal to zero while the rests keep their original values (k). This is done by look at the average of one row and compared to the tolerance.

7.2 Implementing of Baseline Algorithm

- Removing Cov-DL and replace it with a random A
- M is known but not N(k)

7.3 Test on EEG Data Set

To investigate the performance of the main algorithm on real EEG measurements a comparison which the ICA algorithm will be conducted. For this comparison three cases will be investigated: $M = N$, $M < N$ with a third of the sensors removed, $M << N$ with every second sensor removed. All three cases MSE values of the main algorithm will be compared to the same MSE values from the ICA algorithm when $M = N$.

First, a description of how the MSE values from the ICA algorithm is found from the EEG measurement data set. The ICA algorithm take the measurement matrix $\mathbf{Y}_s \in \mathbb{R}^{M \times L_s}$ for each segment s as input and produced a source matrix $\mathbf{X}_s \in \mathbb{R}^{N \times L_s}$ for each segment s. Remember that the number of sensors equal the number of sources, $M = N$, but as mentioned in chapter XX, the case of interest are the active sources k, and by that $N = k$. For each source matrix $\mathbf{X}_s$ one need to find the k active sources but it is not as easy as one may have though. Each entries of the sources matrix are to small to be detected from being active (non-zero) or being non-active (zeros). Instead a tolerance is defined for which values less will be determined as zeros. As the source matrices have positive and negative values a tolerance interval, an interval around zero, must be made. Let the tolerance be defined as tol $= [10E-03, -10E-03]$ where values inside this interval is set equal to zero. A problem occur in form of the stationarity of the sources as described in the motivation chapter **??** sources are stationary if you look at small enough interval. For one second interval this is not the case with our EEG measurements and one can therefore not have a entire row (source) which laid the tolerance interval. One could decrease the length of the segments but one must also take in mind that smaller segments lead to more segments and therefore a higher computational complexity. Instead an average is introduced. For each rows (the sources) of each segments will be average such that one source is resembled by one average value. This average value will then be compared to the interval. If the value laid inside the tolerance

interval, the whole row will be set equal to zero. The sources in each segments equal to zero are removed and the source matrix will now be of size $\mathbf{X}_s \in \mathbb{R}^{k \times L_s}$.

As mentioned only the sensors M is known from the EEG measurement data sets but with the source matrices achieved from the ICA algorithm k is now known for each segment. One now have all the information need to used the main algorithm on the EEG measurement data sets.

7.3.1 Test with $M = N$ on EEG Data Set

Figure 7.1: Visualization of $\text{MSE}(\mathbf{X}, \hat{\mathbf{X}})$ of the main algorithm with simulated stochastic data sets specified by $M = 8$, $L = 1000$ and $k = N$ for $N = M + 1, \cdots, 36$. Average over 500 repetitions for each N.

7.3.2 Test with $\frac{1}{3}M < N$ on EEG Data Set

7.3.3 Test with $\frac{1}{2}M < N$ on EEG Data Set

7.4 Conclusion

Chapter 8

Estimation of Active Sources

Chapter 9

Discussion

Chapter 10

Conclusion

Chapter 11

Further Studies

Bibliography

[1] Aharon, M., Elad, M., and Bruckstein, A. "K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation". In: *IEEE Transactions on signal processing* Vol. 54, No. 11 (2006).

[2] Alickovic, Emina et al. "A Tutorial on Auditory Attention Identification Methods". In: *Front. Neurosci* 13:153 (2019).

[3] Balkan, Ozgur, Kreutz-Delgado, Kenneth, and Makeig, Scott. "Covariance-Domain Dictionary Learning for Overcomplete EEG Source Identification". In: *ArXiv* (2015).

[4] Balkan, Ozgur, Kreutz-Delgado, Kenneth, and Makeig, Scott. "Localization of More Sources Than Sensors via Jointly-Sparse Bayesian Learning". In: *IEEE Signal Processing Letters* (2014).

[5] Balkan, Ozgur Yigit. "Support Recovery and Dictionary Learning for Uncorrelated EEG Sources". Master thesis. University of California, San Diego, 2015.

[6] Bech Christensen, Christian et al. "Toward EEG-Assisted Hearing Aids: Objective Threshold Estimation Based on Ear-EEG in Subjects With Sensorineural Hearing Loss". In: *Trends Hear, SAGE* 22 (2018).

[7] Boyd, S. and Vandenberhe, L. *Convex Optimization*. Cambridge University Press, 2004.

[8] C. Eldar, Yonina and Kutyniok, Gitta. *Compressed Sensing: Theory and Application*. Cambridge University Presse, New York, 2012.

[9] Eaton, Morris L. *Multivariate Statistics: a Vector Space Approach*. John Wiley and Sons, 1983.

[10] Elad, M. *Sparse and Redundant Representations*. Springer, 2010.

[11] Foucart, Simon and Rauhut, Hoger. *A Mathematical Introduction to Compressive Sensing*. Springer Science+Business Media New York, 2013.

[12] Friston, Karl J. "Functional and Effective Connectivity: A Review". In: *Brain Connectivity* 1 (2011).

[13] Friston, Karl J. “Functional integration and inference in the brain”. In: *Progress in Neurobiology 590* 1-31 (2002).

[14] Hyvarinen, A., Karhunen, J., and Oja, E. *Independent Component Analysis*. Ed. by Haykin, Simon. John Wiley and Sons, Inc., 2001.

[15] Makeig, Scott et al. “Blind separation of auditory event-related brain responses into independent components”. In: *Proc. Natl. Acad. Sci. USA* 94 (1997).

[16] Makeig, Scott et al. “Independent Component Analysis of Electroencephalographic Data”. In: *Advances in neural information processing systems* 8 (1996).

[17] Pal, Piya and Vaidyanathan, P. P. “Pushing the Limits of Sparse Support Recovery Using Correlation Information”. In: *IEEE Transactions on Signal Processing* VOL. 63, NO. 3, Feb. (2015).

[18] Palmer, J. A. et al. “Newton Method for the ICA Mixture Model”. In: *ICASSP 2008* (2008).

[19] Sanei, Saeid and Chambers, J.A. *EEG Signal Processing*. Johm Wiley and sons, Ltd, 2007.

[20] Steen, Frederik Van de et al. “Critical Comments on EEG Sensor Space Dynamical Connectivity Analysis”. In: *Brain Topography* 32 p. 643-654 (2019).

[21] *Studies within Steering of hearing devices using EEG and Ear-EEG*. `https://www.eriksholm.com/research/cognitive-hearing-science/eeg-steering`. Accessed: 2019-10-03.

[22] Teplan, M. “Fundamentals of EEG Measurement”. In: *Measurement science review* 2 (2002).

[23] Wipf, D. P. “Bayesian Methods for Finding Sparse Representations”. PhD thesis. University of California, San Diego, 2006.

[24] Wipf, D. P. and Rao, B. D. “An Empirical Bayesian Strategy for Solving the Simultaneous Sparse Approximation Problem”. In: *IEEE Transactions on Signal Processing* Vol. 55.No. 7 (2007).

Appendix A

Extended ICA Algorithms

A.1 Independent Component Analysis

Independent component analysis (ICA) is a method that applies to the general problem of decomposition of a measurement vector into a source vector and a mixing matrix. The intention of ICA is to separate a multivariate signal into statistical independent and non-Gaussian signals and furthermore identify the mixing matrix $\mathbf{A}$, given only the observed measurements $\mathbf{Y}$. A well known application example of source separation is the cocktail party problem, where it is sought to listen to one specific person speaking in a room full of people having interfering conversations. Let $\mathbf{y} \in \mathbb{R}^M$ be a single measurement from M microphones containing a linear mixture of all the speak signals that are present in the room. When additional noise is not considered the problem can be described as the familiar linear system

$$\mathbf{y} = \mathbf{A}\mathbf{x} \tag{A.1}$$

where $\mathbf{x} \in \mathbb{R}^N$ contain the N underlying speak signals and $\mathbf{A}$ is a mixing matrix where the coefficients depends (more or less?) on the distance from the source to the microphone. As such each y_i is a weighted sum of all the present sources of speak.

By ICA both the mixing matrix $\mathbf{A}$ and the source signals $\mathbf{x}$ are sought estimated from the observed measurements $\mathbf{y}$. The main attribute of ICA is the assumption that the sources in $\mathbf{x}$ are statistically independent and non-Gaussian distributed, hence the name independent components.
By independence, one means in general that changes in one source signal do not affect the other source signals. In theory n variables $x_1, \cdots, x_n$ is independent if the joint probability density function (pdf) of $\mathbf{x}$ satisfies

$$p(x_1, x_2, \ldots, x_n) = p_1(x_1)p_2(x_2)\cdots p_n(x_n).$$

The possibility of separating a signal into independent and non-Gaussian components originates from the central limit theorem [14, p. 34]. The theorem states that the distribution of any linear mixture of two or more independent random variables tends toward a Gaussian distribution, under certain conditions.

A.1.1 Assumptions and Preprocessing

For simplicity assume $\mathbf{A}$ is square i.e. $M = N$ and invertible. As such when $\mathbf{A}$ has been estimated the inverse is computed and the components can simply be estimated as $\mathbf{x} = \mathbf{A}^{-1}\mathbf{y}$ [14, p. 152-153].
As both $\mathbf{A}$ and $\mathbf{x}$ are unknown the variances of the independent components can not be determined. However it is reasonable to assume that $\mathbf{x}$ has unit variance – $\mathbf{A}$ is assume to have unit variance as well. Any scalar multiplier within a source can be cancelled out by dividing the corresponding column in $\mathbf{A}$ with the same scalar [14, p. 154]. For further simplification it is assumed without loss of generality that $\mathbb{E}[\mathbf{y}] = 0$ and $\mathbb{E}[\mathbf{x}] = 0$[14, p. 154]. In case this assumption is not true, the measurements can be centred by subtracting the mean as preprocessing before doing ICA.

whiteining is a linear change of coordinates of the mixed data`http://arnauddelorme.com/ica_for_dummies/` "By rotating the axis and minimizing Gaussianity of the projection in the first scatter plot, ICA is able to recover the original sources which are statistically independent

A preprocessing step central to ICA is to whitenthe measurements $\mathbf{y}$. By the whitening process any correlation in the measurements are removed and unit variance is ensured – the independent components $\mathbf{x}$ becomes uncorrelated and have unit variance. Furthermore, this reduces the complexity of ICA and therefore simplifies the recovering process. Whitening is a linear transformation of the observed data. That is multiplying the measurement vector $\mathbf{y}$ with a whitening matrix $\mathbf{V}$,

$$\mathbf{y}_{\text{white}} = \mathbf{V}\mathbf{y}$$

to obtain a new measurement vector $\mathbf{y}_{white}$ that is whited. To obtain a whitening matrix the eigenvalue decomposition (EVD) of the covariance matrix can be used,

$$\mathbb{E}[\mathbf{y}\mathbf{y}^T] = \mathbf{E}\mathbf{D}\mathbf{E}^T$$

where $\mathbf{D}$ is a diagonal matrix of eigenvalues and $\mathbf{E}$ is a matrix consists of the associated eigenvectors. From $\mathbf{E}$ and $\mathbf{D}$ a whitening matrix is constructed [14, p.159].

$$\mathbf{V} = \mathbf{E}\mathbf{D}^{-1/2}\mathbf{E}^T,$$

where $\mathbf{D}^{-1/2} = \text{diag}(d_1^{-1/2}, \ldots, d_n^{-1/2})$ is a componentwise operation.
By multiplying the measurement vector $\mathbf{y}$ with a whitening matrix $\mathbf{V}$ the data becomes white

$$\mathbf{y}_{\text{white}} = \mathbf{V}\mathbf{y} = \mathbf{V}\mathbf{A}\mathbf{x} = \mathbf{A}_{\text{white}}\mathbf{x}.$$

Furthermore the mixing matrix $\mathbf{A}_{\text{white}}$ becomes orthogonal

$$\mathbb{E}[\mathbf{y}_{\text{white}}\mathbf{y}_{\text{white}}^T] = \mathbf{A}_{\text{white}}\mathbb{E}[\mathbf{x}\mathbf{x}^T]\mathbf{A}_{\text{white}}^T = \mathbf{A}_{\text{white}}\mathbf{A}_{\text{white}}^T = \mathbf{I},$$

where $\mathbb{E}[\mathbf{x}\mathbf{x}^T] = \mathbf{I}$ because of assumed uncorrelation and zero mean. Consequently ICA can restrict its search for the mixing matrix to the orthogonal matrix space – that is instead of estimating N^2 parameters ICA one now only has to estimate an orthogonal matrix which has $N(N-1)/2$ parameters/degrees of freedom [14, p. 159] .

se udkommentering herunder?

A.1.2 Recovery of the Independent Components

Now the ICA model is established, the next step is the estimation of the mixing coefficients a_{ij} and independent components x_i. The simple and intuitive method is to take advantage of the assumption of non-Gaussian independent components. Consider again the ICA model of a single measurement vector $\mathbf{y} = \mathbf{A}\mathbf{x}$ where the independent components can be estimated by the inverted model $\mathbf{x} = \mathbf{A}^{-1}\mathbf{y}$. Let $\mathbf{A}^{-1} = \mathbf{B}$, now a single independent component can be seen as the linear combination

Herfra og ned til kurtosis skal lige tjekkes i gennem og rettes til

$$x_i = \mathbf{b}_{i\cdot}\mathbf{y} = \sum_k b_k y_k \tag{A.2}$$

where $\mathbf{b}_{i\cdot}$ is the i-th row of $\mathbf{B}$. The issue is now to determine $\mathbf{b}_{i\cdot}$ such that it equals the i-th row from the inverse $\mathbf{A}$, as it is assumed. As $\mathbf{A}$ is unknown it is not possible to determine $\mathbf{b}_{i\cdot}$ exactly, but an estimate can be found to make a good approximation. Rewriting (**??**)

$$x_i = \mathbf{b}_{i\cdot}\mathbf{y} = \mathbf{b}_{i\cdot}\mathbf{A}\mathbf{x} = \mathbf{q}^T\mathbf{x} = \sum_{k=1} q_k x_k$$

it is seen how x_i is a linear combination of all x_k, thus the equality only holds true when $\mathbf{q}$ consist of only one non-zero element that equals 1. Due to the central limit theorem the distribution of $\mathbf{q}^T\mathbf{x}$ is most non-Gaussian when it equals one of the independent components which was assumed non-Gaussian. Then, since $\mathbf{q}^T\mathbf{x} = \mathbf{b}_{i\cdot}\mathbf{y}$, it is possible to vary the coefficients in $\mathbf{b}$ and look at the distribution of $\mathbf{b}_{i\cdot}\mathbf{y}$. Finding the vector $\mathbf{b}_{i\cdot}^T$ that maximizes the non-Gaussianity would then correspond to $\mathbf{q} = \mathbf{A}^T\mathbf{b}_{i\cdot}^T$ having only a single non-zero element. Thus maximizing the non-Gaussianity of $\mathbf{b}_{i\cdot}\mathbf{y}$ results in one of the independent components [14, p. 166].
Considering the N-dimensional space of vectors $\mathbf{b}_{i\cdot}^T$ there exist $2N$ local maxima, corresponding to x_i and $-x_i$ for all n independent components [14, p. 166].

A.1.3 Kurtosis

To maximize the non-Gaussianity a measure for Gaussianity is needed. Kurtosis is a quantitative measure used for non-Gaussianity of random variables. Kurtosis of a

random variable y is the fourth-order cumulant denoted by $\text{kurt}(y)$. For y with zero mean and unit variance, kurtosis reduces to

uddyb? og tilføj k kurtosis

$$\text{kurt}(y) = \mathbb{E}[y^4] - 3.$$

It is seen that the kurtosis is a normalized version of the fourth-order moment defined as $\mathbb{E}[y^4]$. For a Gaussian random variable the fourth-order moment equals $3(\mathbb{E}[y^2])^2$ hence the corresponding kurtosis will be zero [14, p. 171]. Consequently the kurtosis of non-Gaussian random variables will almost always be different from zero.
The kurtosis is a common measure for non-Gaussianity due to its simplicity both theoretical and computational. The kurtosis can be estimated computationally by the fourth-order moment of sample data when the variance is constant. Furthermore, for two independent random variables x_1, x_2 the following linear properties applies to the kurtosis of the sum

$$\text{kurt}(x_1 + x_2) = \text{kurt}(x_1) + \text{kurt}(x_2) \quad \text{and} \quad \text{kurt}(\alpha x_1) = \alpha^4 \text{kurt}(x_1)$$

However, one complication concerning kurtosis as a measure is that kurtosis is sensitive to outliers [14, p. 182].

Consider again the vector $\mathbf{q} = \mathbf{A}^T\mathbf{b}$ such that $\mathbf{b}_{i\cdot}\mathbf{y} = \sum_{k=1} q_k x_k$. By the additive property of kurtosis

$$\text{kurt}\left(\mathbf{b}_{i\cdot}\mathbf{y}\right) = \sum_{k=1} q_k^4 \text{kurt}(x_k).$$

hvordan kommer dette frem?

Then the assumption of the independent components having unit variance results in $\mathbb{E}[x_i^2] = \sum_{k=1} q_k^2 = 1$. That is geometrically that $\mathbf{q}$ is constrained to the unit sphere, $\|\mathbf{q}\|^2 = 1$. By this the optimisation problem of maximising the kurtosis of $\mathbf{b}_{i\cdot}\mathbf{y}$ is similar to maximizing $|\text{kurt}(x_i)| = |\sum_{k=1} q_k^4 \text{kurt}(x_k)|$ on the unit sphere.
Due to the described preprocessing $\mathbf{b}_{i\cdot}^T$ is assumed to be white and it can be shown that $\|\mathbf{q}\| = \|\mathbf{b}_{i\cdot}^T\|$ [14, p. 174]. This shows that constraining $\|\mathbf{q}\|$ to one is similar to constraining $\|\mathbf{b}_{i\cdot}^T\|$ to one.

A.1.4 The Gradient Algorithm with Kurtosis

In practise, to recover the mixing matrix $\mathbf{A}$ by maximizing the kurtosis of $\mathbf{b}_{i\cdot}\mathbf{y}$, gradient optimisation methods are used.
The general idea behind a gradient algorithm is to determine the direction for which $\text{kurt}(\mathbf{b}_{i\cdot}\mathbf{y})$ is growing the most, based on the gradient.
The gradient of $|\text{kurt}(\mathbf{b}_{i\cdot}\mathbf{y})|$ is computed as

$$\frac{\partial|\text{kurt}(\mathbf{b}_{i\cdot}\mathbf{y})|}{\partial \mathbf{b}_j} = 4\text{sign}(\text{kurt}(\mathbf{b}_{i\cdot}\mathbf{y}))(\mathbb{E}[\mathbf{y}(\mathbf{b}_{i\cdot}\mathbf{y})^3] - 3\mathbf{y}\mathbb{E}[(\mathbf{b}_{i\cdot}\mathbf{y})^2]) \tag{A.3}$$

As $\mathbb{E}[(\mathbf{b}_{i\cdot}\mathbf{y})^2] = \|\mathbf{y}\|^2$ for whitened data the corresponding term does only affect the norm of $\mathbf{b}_{i\cdot}$ within the gradient algorithm. Thus, as it is only the direction that is of interest, this term can be omitted. Because the optimisation is restricted to the unit sphere a projection of $\mathbf{b}_{i\cdot}^T$ onto the unit sphere must be performed in every step of the gradient method. This is done by dividing $\mathbf{b}_{i\cdot}^T$ by its norm. This gives update step

$$\begin{aligned}\Delta\mathbf{b}_{i\cdot}^T &\propto \operatorname{sign}\left(\operatorname{kurt}(\mathbf{b}_{i\cdot}\mathbf{y})\right)\mathbb{E}[\mathbf{y}(\mathbf{b}_{i\cdot}\mathbf{y})^3] \\ \mathbf{b}_{i\cdot}^T &\leftarrow \mathbf{b}_{i\cdot}^T/\|\mathbf{b}_{i\cdot}^T\|\end{aligned}$$

The expectation operator can be omitted in order to achieve an adaptive version of the algorithm, now using every measurement $\mathbf{y}$. However, the expectation operator from the definition of kurtosis can not be omitted and must therefore be estimated. This can be done by γ by serving it as the learning rate of the gradient method.

$$\Delta\gamma \propto ((\mathbf{b}_{i\cdot}\mathbf{y})^4 - 3) - \gamma$$

A.1.5 Basic ICA algorithm

Algorithm **??** combines the above theory, to give an overview of the ICA procedure.

Algorithm 3 Basis ICA

procedure PRE-PROCESSING($\mathbf{y}$)
  Center measurements $\mathbf{y} \leftarrow \mathbf{y} - \bar{\mathbf{y}}$
  Whitening $\mathbf{y} \leftarrow \mathbf{y}_{white}$
end procedure

procedure ICA($\mathbf{y}$)
  $k = 0$
  Initialise random vector $\mathbf{b}_{i\cdot(k)}$ ▷ unit norm
  Initialise random value $\gamma_{(k)}$
  for $j \leftarrow 1, 2, \cdots, N$ **do**
    while convergence critia not meet **do**
      $k = k + 1$
      $\mathbf{b}_{i\cdot(k)} \leftarrow \operatorname{sign}\gamma_{(k-1)}\mathbf{y}(\mathbf{b}_{i\cdot}\mathbf{y})^3$
      $\mathbf{b}_{i\cdot(k)} \leftarrow \mathbf{b}_{i\cdot(k)}/\|\mathbf{b}_{i\cdot(k)}\|$
      $\gamma_{(k)} \leftarrow ((\mathbf{b}_{i\cdot}\mathbf{y})^4 - 3) - \gamma_{(k-1)}$
    end while
    $x_j = \mathbf{b}_{i\cdot}\mathbf{y}$
  end for
end procedure

A.1.6 ICA for sparse signal recovery

ICA is widely used within sparse signal recovery. When ICA is applied to a measurement vector $\mathbf{y} \in \mathbb{R}^M$ it is possible to separate the mixed signal into M or less independent components. However, by assuming that the independent components make a k-sparse signal it is possible to apply ICA within sparse signal recovery of cases where $M < N$ and $k \leq M$.
To apply ICA to such cases the independent components are obtained by the pseudo-inverse solution

$$\hat{\mathbf{x}} = \mathbf{A}_S^{\dagger}\mathbf{y}$$

where $\mathbf{A}_S$ is derived from the dictionary matrix $\mathbf{A}$ by containing only the columns associated with the non-zero entries of $\mathbf{x}$, specified by the support set S.

henvis til appendix

This appendix provide an extension to the basic algorithm for ICA regarding the measure of non-Gaussianity and the computation method. This extended algorithm is referred to as fast ICA and is more commonly used for source separation. This is the algorithm used to apply ICA on EEG measurements for comparison within the thesis.

A.2 Fixed-Point Algorithm - FastICA

An advantage of gradient algorithms is the possibility of fast adoption in non-stationary environments due the use of all input, $\mathbf{y}$, at once. A disadvantage of the gradient algorithm is the resulting slow convergence, depending on the choice of γ for which a bad choice in practise can disable convergence. A fixed-point iteration algorithm to maximise the non-Gaussianity is an alternative that could be used.
Consider the gradient step derived in section **??**. In the fixed point iteration the sequence of γ is omitted and replaced by a constant. This builds upon the fact that for a stable pointof the gradient algorithm the gradient must point in the direction of $\mathbf{b}_j$, hence be equal to $\mathbf{b}_j$. In this case adding the gradient to $\mathbf{b}_j$ does not change the direction and convergence is achieved.

wiki: The fixed point is stable if the absolute value of the derivative of **w** at the point is strictly less than 1?

Letting the gradient given in (**??**) be equal to $\mathbf{w}$ and considering the same simplifications again suggests the new update step as [14, p. 179]

$$\mathbf{b}_j \leftarrow \mathbb{E}[\mathbf{y}(\mathbf{b}_j^T\mathbf{y})^3] - 3\mathbf{b}_j.$$

After the fixed point iteration $\mathbf{b}_j$ is again divided by its norm to withhold the constraint $\|\mathbf{b}_j\| = 1$. Instead of γ the fixed-point algorithm compute $\mathbf{b}_j$ directly from previous $\mathbf{b}_j$.
The fixed-point algorithm is referred to as FastICA. The algorithm has shown to converge fast and reliably, then the current and previous $\mathbf{w}$ laid in the same direction [14, p. 179].

A.2.1 Negentropy

An alternative measure of non-Gaussianity is the negentropy, which is based on the differential entropy. The differential entropy H of a random vector $\mathbf{y}$ with density $p_y(\boldsymbol{\eta})$ is defined as

$$H(\mathbf{y}) = -\int p_y(\boldsymbol{\eta}) \log(p_y(\boldsymbol{\eta}))\ d\boldsymbol{\eta}.$$

The entropy describes the information that a random variable gives. The more unpredictable and unstructured a random variable is higher is the entropy, e.g. Gaussian random variables have a high entropy, in fact the highest entropy among the random variables of the same variance [14, p. 182].

Negentropy is a normalised version of the differential entropy such that the measure of non-Gaussianity is zero when the random variable is Gaussian and non-negative otherwise. The negentropy J of a random vector $\mathbf{y}$ is defined as

$$J(\mathbf{y}) = H(\mathbf{y}_{\text{gaus}}) - H(\mathbf{y}),$$

with $\mathbf{y}_{\text{gaus}}$ being a Gaussian random variable of the same covariance and correlation as $\mathbf{y}$ [14, p. 182].

As the kurtosis is sensitive for outliers the negentropy is instead difficult to compute computationally as the negentropy require a estimate of the pdf. As such an approximation of the negentropy is needed.

To approximate the negentropy it is common to use the higher order comulants including the kurtosis. The following approximation is stated without further elaboration, the derivation can be found in [14, p. 182].

A.2.2 Fixed-Point Algorithm with Negentropy

Maximization of negentropy by use of the fixed-point algorithm is now presented, for derivation of the fixed point iteration see [14, p. 188]. Algorithm **??** show Fast ICA using negentropy, this is the algorithm which is implemented for comparison with the source separation methods which are tested in this thesis.

Algorithm 4 Fast ICA – with negentropy

procedure PRE-PROCESSING($\mathbf{y}$)
    Center measurements $\mathbf{y} \leftarrow \mathbf{y} - \bar{\mathbf{y}}$
    Whitening $\mathbf{y} \leftarrow \mathbf{y}_{white}$
end procedure

procedure FASTICA($\mathbf{y}$)
    $k = 0$
    Initialise random vector $\mathbf{b}_{j(k)}$ ▷ unit norm
    for $j \leftarrow 1, 2, \cdots, N$ **do**
        while convergance critia not meet **do**
            $k = k + 1$
            $\mathbf{b}_{j(k)} \leftarrow \mathbb{E}[\mathbf{y}(\mathbf{b}_j^T\mathbf{y})] - \mathbb{E}[g'(\mathbf{b}_j^T\mathbf{y})]\mathbf{b}_j$ ▷ g defined in [14, p. 190]
            $\mathbf{b}_{j(k)} \leftarrow \mathbf{b}_j / \|\mathbf{b}_j\|$
        end while
        $x_j = \mathbf{b}_j^T\mathbf{y}$
    end for
end procedure

A.3 Verification of fast ICA on synthetic data

The purpose of this section is to verify the fast ICA algorithm which is used in this thesis. By this verification the purpose is to justify the ICA algorithm as a reference point with respect to performance of the developed main algorithm.

The fast ICA algorithm is tested on synthetic data simulated as described in section 6.2. Consider the following linear system, which makes a model of EEG measurements.

$$\mathbf{Y} = \mathbf{A}\mathbf{X}$$

where $\mathbf{Y}^{M\times L}$, $\mathbf{A}^{M\times N}$ and $\mathbf{x}^{N\times L}$. It is expected that the fast ICA algorithm manage to solve the linear system for $\mathbf{X}$ given only the measurements $\mathbf{Y}$, in the case where $M = N$. A

Furthermore it is of interest to determine $\mathbf{A}$ from the fast ICA algorithm.

Appendix B

Supplementary theory for chapter 4

description of content of the chapter.

B.1 Introduction to Compressive Sensing

Compressive sensing is the theory of efficient recovery of a signal from a minimal number of observed measurements. It is build upon empirical observations assuring that many signals can be approximated by remarkably sparser signals. Assume linear acquisition of the observed measurements, then the relation between the measurements and the signal to be recovered can be modelled by the multiple measurement vector (MMV) model (3.2) [11].

Through this section the introduction of the theory behind compressive sensing will be presented for one measurement vector of (3.2), $\mathbf{y}$, such that the theory is based on the linear system (3.1). This will be done for simplicity but the theory will still apply for the extend linear system (3.2).

In compressive sensing terminology, $\mathbf{x} \in \mathbb{R}^N$ is the signal of interest which is sought recovered from the EEG measurement $\mathbf{y} \in \mathbb{R}^M$ by solving the linear system (3.1). In the typical compressive sensing case the system is under-determined, $M < N$, and there will therefore exist infinitely many solutions, provided that one solution exist. However, by enforcing certain sparsity constraints it is possible to recover the wanted signal, hence the term sparse signal recovery [11]. The sparsity constraints are the ones presented in 3.1 where the ℓ_0 is introduced to count the non-zeros of the signal of interest, the source vector $\mathbf{x}$. The number of non-zeros (active sources) k describe how sparse the source vector is.

To find a k-sparse solution to the linear system (3.1) it can be viewed as the

following optimisation problem.

$$\mathbf{x}^* = \arg\min_{\mathbf{x}\in\mathbb{C}} \|\mathbf{x}\|_0 \quad \text{subject to} \quad \mathbf{A}\mathbf{x} = \mathbf{y}.$$

Unfortunately, this optimisation problem is non-convex due to the definition of the ℓ_0-norm and is therefore difficult to solve – it is an NP-hard problem. Instead, by replacing the ℓ_0-norm with the ℓ_1-norm, the optimisation problem can be approximated and hence becomes computationally feasible [8, p. 27]

4.1: skal vi indfør z som en approximation til x. og så et nyt omega eller? eller kan vi lade x* være løslingen til både P0 og P1

$$\mathbf{x}^* = \arg\min_{\mathbf{x}\in\mathbb{C}} \|\mathbf{x}\|_1 \quad \text{subject to} \quad \mathbf{A}\mathbf{x} = \mathbf{y}. \tag{B.1}$$

With this optimisation problem the best k-sparse solution $\mathbf{x}^*$ can be found. The optimisation problem is referred to as ℓ_1 optimisation problem or Basis Pursuit. The following theorem justifies that the ℓ_1 optimisation problem finds a sparse solution [11, p. 62-63].

Theorem B.1.1
A mixing matrix $\mathbf{A} \in \mathbb{R}^{M\times N}$ is defined with columns $\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_N]$. By assuming uniqueness of a solution $\mathbf{x}^*$ to

$$\min_{\mathbf{x}\in\mathbb{R}^N} \|\mathbf{x}\|_1 \quad \text{subject to} \quad \mathbf{A}\mathbf{x} = \mathbf{y},$$

the system $\{\mathbf{a}_j, j \in \text{supp}(\mathbf{x}^*)\}$ is linearly independent, and in particular

$$\|\mathbf{x}^*\|_0 = \text{card}(\text{supp}(\mathbf{x}^*)) \leq M.$$

Proof
Assume that the set $\{\mathbf{a}_l, l \in S\}$ of l columns from matrix $\mathbf{A} \in \mathbb{R}^{M\times N}$ is linearly dependent with the support $S = \text{supp}(\mathbf{x}^*)$. Thus a non-zero vector $\mathbf{v} \in \mathbb{R}^N$ supported on S exists such that $\mathbf{A}\mathbf{v} = \mathbf{0}$ – the system is linear dependent. The unique solution $\mathbf{x}^*$ can then be written as, for any $t \neq 0$,

$$\|\mathbf{x}^*\|_1 < \|\mathbf{x}^* + t\mathbf{v}\|_1 = \sum_{l\in S} |x_l^* + tv_l| = \sum_{l\in S} \text{sgn}(x_l^* + tv_l)(x_l^* + tv_l). \tag{B.2}$$

For a small $|t|$

$$|t| < \min_{l\in S} \frac{|x_l^*|}{\|\mathbf{v}\|_\infty},$$

then the sign function become

$$\text{sgn}(x_l^* + tv_l) = \text{sgn}(x_l^*), \quad \forall l \in S.$$

By including this result in (**??**) and remembering $t \neq 0$:

$$\|\mathbf{x}^*\|_1 < \sum_{l \in S} \operatorname{sgn}(x_l^*)(x_l^* + t v_l) = \sum_{l \in S} \operatorname{sgn}(x_l^*) x_l^* + t \sum_{l \in S} \operatorname{sgn}(x_l^*) v_l = \|\mathbf{x}^*\|_1 + t \sum_{l \in S} \operatorname{sgn}(x_l^*) v_l.$$

From this it can be seen that it is always possible to choose $t \neq 0$ small enough such that

$$t \sum_{l \in S} \operatorname{sgn}(x_l^*) v_l \leq 0,$$

which contradicts that $\mathbf{v}$ make the columns of $\mathbf{A}$ linear dependent. Therefore, the set $\{\mathbf{a}_l, l \in S\}$ must be linearly independent. ■

From the theorem is must be conclude that the choice of the mixing matrix $\mathbf{A}$ has a significant impact on whenever a unique solution $\mathbf{x}^*$ exist for the ℓ_1 optimisation problem (**??**). Therefore, when recovering $\mathbf{A}$, some considerations regarding the recovering process of $\mathbf{A}$ must be taken into account. A method for the recovering of $\mathbf{A}$ could be to use a dictionary. This will be explain in the following section 4.2.1.

An alternative solution method to the ℓ_1 optimisation includes greedy algorithms such as the Orthogonal Matching Pursuit (OMP) [11, P. 65]. The OMP algorithm is an iteration process where an index set S is updated – at each iteration – by adding indices corresponding to the columns of $\mathbf{A}$ which describe the residual best possible, hence greedy. The vector $\mathbf{x}$ is then updated by a vector supported on S which minimise the residual, that is the orthogonal projection of $\mathbf{y}$ onto the $\operatorname{span}\{\mathbf{a}_l \mid l \in S\}$.

B.2 K-SVD Algorithm

The dictionary learning algorithm K-SVD provides an updating rule which is applied to each column of $\mathbf{A}_0 = [\mathbf{a}_0, \ldots, \mathbf{a}_N]$ where $\mathbf{A}_0$ being a random initial dictionary matrix. Updating first $\mathbf{a}_j$ and then the corresponding coefficients in $\mathbf{X}$ which it is multiplied with the i-th row in $\mathbf{X}$ denoted by $\mathbf{x}_{i\cdot}$. Let $\mathbf{a}_{j_0}$ be the column to be updated and let the remaining columns be fixed. By rewriting the objective function in (4.4) using matrix notation it is possible to isolate the contribution from $\mathbf{a}_{j_0}$.

Tjek nedenstående udledning. a og x er ikke lige lange da a_j er M lang mens $x_{i\cdot}$ er L lang

$$\begin{aligned} \|\mathbf{Y} - \mathbf{A}\mathbf{X}\|_F^2 &= \left\| \mathbf{Y} - \sum_{j=1}^{N} \mathbf{a}_j \mathbf{x}_{i\cdot} \right\|_F^2 \\ &= \left\| \left(\mathbf{Y} - \sum_{j \neq j_0}^{N} \mathbf{a}_j \mathbf{x}_{i\cdot} \right) - \mathbf{a}_{j_0} \mathbf{x}_{i_0\cdot} \right\|_F^2, \end{aligned} \tag{B.3}$$

where $i = j$, $i_0 = j_0$ and where F is the Frobenius norm that works on matrices

$$\|\mathbf{A}\|_F = \sqrt{\sum_{i=1}^{M} \sum_{j=1}^{N} |a_{ij}|^2}.$$

In (**??**) the term in the parenthesis is denoted by $\mathbf{E}_{j_0}$, an error matrix, and hence by minimising (**??**) with respect to $\mathbf{a}_{j_0}$ and $\mathbf{x}_{i_0\cdot}$ leads to the optimal contribution from j_0

$$\min_{\mathbf{a}_{j_0},\mathbf{x}_{i_0\cdot}} \| \mathbf{E}_{j_0} - \mathbf{a}_{j_0}\mathbf{x}_{i_0\cdot}\|_F^2 . \tag{B.4}$$

The optimal solution to (**??**) is known to be the rank-1 approximation of $\mathbf{E}_{j_0}$. This comes from the Eckart–Young–Mirsky theorem [**?**] saying that a partial single value decomposition (SVD) makes the best low-rank approximation of a matrix such as $\mathbf{E}_{j_0}$. The SVD is given as

$$\mathbf{E}_{j_0} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \in \mathbb{R}^{M\times N},$$

with $\mathbf{U} \in \mathbb{R}^{M\times M}$ and $\mathbf{V} \in \mathbb{R}^{N\times N}$ being unitary matrices[1] and $\mathbf{\Sigma} = \text{diag}[\boldsymbol{\sigma}_1, \ldots, \boldsymbol{\sigma}_M] \in \mathbb{R}^{M\times N}$ a diagonal matrix. $\boldsymbol{\sigma}_j$ are the non-negative singular values of $\mathbf{E}_{j_0}$. The best k-rank approximation to $\mathbf{E}_{j_0}$, with $k < \text{rank}(\mathbf{E}_{j_0})$ is then given by :

kilde

$$\mathbf{E}_{j_0}^{(k)} = \sum_{j=1}^{k} \boldsymbol{\sigma}_j \mathbf{u}_j \mathbf{v}_j^T .$$

Since the outer product always have rank-1 letting $\mathbf{a}_{j_0} = \mathbf{u}_1$ and $\mathbf{x}_{i_0\cdot} = \boldsymbol{\sigma}_j \mathbf{v}_1^T$ solves the optimisation problem (**??**). However in order to preserve the sparsity in $\mathbf{X}$ while optimising, only the non-zero entries in $\mathbf{x}_{i_0\cdot}$ are allowed to vary. For this purpose only a subset of columns in $\mathbf{E}_{j_0}$ is considered, those which correspond to the non-zero entries of $\mathbf{x}_{i_0\cdot}$. A matrix $\mathbf{P}_{i_0}$ is defined to restrict $\mathbf{x}_{i_0\cdot}$ to only contain the non-zero-rows corresponding to N_{j_0} non-zero rows:

$$\mathbf{x}_{i_0\cdot}^{(R)} = \mathbf{x}_{i_0\cdot}\mathbf{P}_{i_0}$$

where R denoted the restriction. By applying the SVD to the error matrix which has been restricted $\mathbf{E}_{j_0}^{(R)} = \mathbf{E}_{j_0}\mathbf{P}_{i_0}$ and updating $\mathbf{a}_{j_0}$ and $\mathbf{x}_{i_0\cdot}^{(R)}$ the rank-1 approximation is found and the original representation vector is updated as $\mathbf{x}_{i_0\cdot} = \mathbf{x}_{i_0\cdot}^{(R)}\mathbf{P}_{i_0}^T$.

The main steps of K-SVD is described in algorithm **??**.

[1] Unitary matrix: $\mathbf{U}^T\mathbf{U} = \mathbf{U}\mathbf{U}^T = \mathbf{I}$

Algorithm 5 K-SVD

$k = 0$
Initialize random $\mathbf{A}_{(0)}$
Initialize $\mathbf{X}_{(0)} = \mathbf{0}$

procedure K-SVD($\mathbf{A}_{(0)}$)
    Normalize columns of $\mathbf{A}_{(0)}$
    while error $\geq$ limit **do**
        $j = j + 1$
        for $j \leftarrow 1, 2, \ldots, L$ **do** ▷ updating each col. in $\mathbf{X}_{(k)}$
            $\hat{\mathbf{x}}_j = \min_{\mathbf{x}} \|\mathbf{y}_j - \mathbf{A}_{(k-1)}\mathbf{x}_j\|$ subject to $\|\mathbf{x}_j\| \leq k$ ▷ use Basis Pursuit
        end for
        $\mathbf{X}_{(k)} = \{\hat{\mathbf{x}}_j\}_{j=1}^{L}$
        for $j_0 \leftarrow 1, 2, \cdots, N$ **do**
            $\Omega_{j_0} = \{j \mid 1 \leq j \leq L, \mathbf{X}_{(k)}[j_0, j] \neq 0\}$
            From Ω_{j_0} define $\mathbf{P}_{i_0}$
            $\mathbf{E}_{j_0} = \mathbf{Y} - \sum_{j \neq j_0}^{N} \mathbf{a}_j \mathbf{x}_{i\cdot}$
            $\mathbf{E}_{j_0}^{(R)} = \mathbf{E}_{j_0}\mathbf{P}_{i_0}$
            $\mathbf{E}_{j_0}^{(R)} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$ ▷ perform SVD
            $\mathbf{a}_{j_0} \leftarrow \mathbf{u}_1$ ▷ update the j_0 col. in $\mathbf{A}_{(k)}$
            $(\mathbf{x}_{i_0\cdot})^{(R)} \leftarrow \boldsymbol{\sigma}_1 \mathbf{v}_1$
            $\mathbf{x}_{i_0\cdot} \leftarrow (\mathbf{x}_{i_0\cdot})^{(R)} \mathbf{P}_{i_0}^T$ ▷ update the i_0 row in $\mathbf{X}_{(k)}$
        end for
        error = $\|\mathbf{Y} - \mathbf{A}_{(k)}\mathbf{X}_{(k)}\|_F^2$
    end while
end procedure

B.3 Principal Component Analysis

B.4 General Optimation Theory...or more specific what?

Appendix C

List of Scripts

.. remember to setup